ST/CES/9

CONFERENCE OF EUROPEAN STATISTICIANS

STATISTICAL STANDARDS AND STUDIES ---------- No. 9

AUTOMATIC FILES IN STATISTICAL SYSTEMS

UNIITED NATIONS

THE CONFERENCE OF EUROPEAN STATISTICIANS

The Conference of European Statisticians was set up in 1953 as a continuing body meeting under the auspices of the United Nations. Its objectives are (a) to improve European official statistics and their international comparability having regard to the recommendations of the Statistical Commission of the United Nations, the Specialized Agencies and other appropriate bodies as necessary; and (b) to promote close co-ordination of the statistical activities in Europe of international organizations so as to achieve greater uniformity in concepts and definitions and to reduce to a minimum the burdens on national statistical offices. The members to the Conference are the directors of the central statistical offices of the countries participating in the work of the United Nations Economic Commission for Europe. The Conference meets in plenary session once a year and also arranges numerous meetings of specialists on particular statistical subjects.

UNITED NATIONS STATISTICAL COMMISSION and
ECONOMIC COMMISSION FOR EUROPE

CONFERENCE OF EUROPEAN STATISTICIANS

STATISTICAL STANDARDS AND STUDIES ---- No.9

AUTOMATIC FILES IN STATISTICAL SYSTEMS

UNITED NATIONS

New York, 1967

ST/CES/9

UNITED NATIONS PUBLICATION

Sales Number : 67.II.E/Mim.41

Preface

The Conference of European Statisticians, at its thirteenth plenary session in October 1965, agreed that its Working Group on Electronic Data Procession (EBP) should be convened to consider the subject of automatic files in statistical systems. As a basis for this discussion, the Secretariat arranged to have a paper prepared by a consultant, Mr. Svein Nordbotten of the Central Bureau of Statistics of Norway.

The Working Group studied this paper at a meeting in March 1967, which was attended by participants from Austria, Belgium, Bulgaria, Byelorussian SSH, Canada, Czechoslovakia, Denmark, Federal Republic of Germany, France, Greece, Hungary, Ireland, Italy, Luxembourg, Netherlands, Norway, Poland, Romania, Spain, Sweden, Switzerland, Union of Soviet Socialist Republics, United Kingdom, United States and Yugoslavia as well as representatives of several international organizations.

The Working Group agreed that the paper made a significant contribution to the study of an important set of problems currently facing many national statistical offices, namely how to organise the national statistical work, including collection operations, so that the potentialities of automatic data processing equipment are fully exploited. It was noted that the paper dealt only with the file aspects of this problem; also it did not offer ready-made solutions but was intended to form a basis of further discussions, both national and international. The Working Group recommended that, in view of the general interest of the issues studied, the paper should be revised in the light of the Group's discussion and published for the benefit of a wider audience.

While this material has the general approval of the experts who participated in the Working Group, it is issued in the name of the consultant and the Secretariat.

AUTOMATIC FILES IN STATISTICAL SYSTEMS

by

Mr. Svein Nordbotten

Central Bureau of Statistics, Norway

(Consultant to the Secretariat of the Conference of European Statisticians)

Table of Contents

1. Introduction

The existing statistical systems are designed for satisfying conditions which are changing rapidly today. The most marked feature of the existing systems of many countries is their extensive periodic surveys.

The surveys aim at presentation of statistics which are internally consistent and compatible with results for different periods or points of time. The individual observations of any two surveys are, however, usually incompatible because the individuals are not permanently identified. This individual incompatibility implies that any survey must be self-supplying with respect to all data required for the compatibility of aggregates. The system requires that respondents have to answer certain questions about facts reported one or perhaps many times previously because these are needed for classification purposes in order to obtain results compatible with those from preceding or parallel surveys.

The data collected for a survey have usually been utilized in one computation process and have been considered as useless for any later utilization because of the large cost of making them ready for a new process. This has led to extensive processing schemes for each survey aiming at a large volume of general purpose results for present and anticipated needs. This means totals or averages for a large number of classes in the population investigated. The extensive schemes of processing have also been very time-consuming with statistical results which, for example, were available 2-4 years after a decennial census was taken. The average age for statistics from such censuses would therefore be 7-9 years.

What are the future needs for statistical information from national statistical systems in societies which are growing more and more complex? A probable answer to the question may perhaps be: special 'statistical services and fast statistical measurement of changes in addition to the general purpose statistics.

The statistical aggregates available at present are well suited for explaining the behavior or technical manner of operation of large groups in macro analyses. However, it is not usually these groups which behave or operate, but their individual members. The aggregate will, moreover, hide internal individual variation which may be of greater analytical value than variation between aggregates.

To develop more basic and useful explanations, the analysts will have to ask for special computations on the individual observations.

Higher productivity, faster production, new means of communication and transportation, inter-dependence, etc. of modern countries require fast and correct decisions, based on up-to-date statistical information about changes and state of affairs.

This may be obtained by several means. Using data collected for other non-statistical purposes more efficiently, may be one way. A second way may be to collect data for samples of units each year instead of taking censuses of all units every tenth year. A third way would be to work out estimates of the present situation, i.e. forecasts of already realized but unobserved situations, on a current basis utilizing intensively available data from many sources about individual units in the estimation process.

All the above proposals for satisfying future needs for statistical information assume that individual observations can be preserved, linked together and used when required without prohibitive costs. This may be implemented by means of a data file thanks to the development of the electronic data processing equipment [**16**].

Modern data processing systems may contribute to this implementation of data files in two ways. First, they represent an efficient automatic tool for data storage and handling which makes it possible to work with millions of data. Second, the application of data processing machines in an increasing number of administrative processes in government agencies makes the acquisition of data collected for non-statistical purposes easier and less expensive.

Ideas about central data files have been promoted in many connections and by different people. Researchers in economics and social sciences have proposed, discussed and established automatic files for data about macro units compiled from statistical publications "as well as unlinked data on micro units from special sample surveys. Several national statistical offices have also considered the possibility of establishing central data files with data for both macro and micro units identified in such a

way that linkage is possible [**7, 8, 19, 20, 21, 22**].

In this paper we shall discuss some of the problems connected with such central data files from the point of view of a central statistical office and with special reference to the data processing aspects. Even though the establishment of a data file may have important effects on data collection techniques and computation of statistics, these consequences are not considered within the scope of this paper.

2. Statistical systems

2.1 General model

We shall call the statistical system we are going to discuss a statistical file system in contrast to the traditional statistical system. It may be useful to start the discussion with a few theoretical considerations [**18**].

We assume that the size of the national product is partly dependent on the knowledge incorporated in the society. The knowledge is not consumed in the process of production, but is made use of more like the way production capital is used. A special part of the knowledge is of statistical nature, i.e. it is computed from a set of collected observations. (See **Figure 1**.)

The supply of statistical information to the society takes place by the publication of the computed results. The supply is determined by two factors, the available stock of computed statistics and the degree of utilization and multiplication of these in publications. Let this process be denoted by:

(1) I = I (M, S)

where I is the supply of information per time unit at a point of time, **S** is the stock of computed statistics and **M** represents the degree of utilization and multiplication of **S** in publications, both referring to the same point of time as **I**.

We shall call **S** the computed data capital because the computed statistics are data in the process of supplying information and because they participate in this process just in the same way as physical capital in a production process. Statistical information may therefore be supplied without any change in the computed capital, e.g. by issuing new editions of statistical publications or publishing new compositions of previously computed statistics.

The computed capital is increased by investing. Investment in computed capital is done by computing new statistics either from collected data or from previously computed results. The investment must therefore partly be determined by available stocks of previously computed results and of collected data, and partly by the degree of utilization of these two stocks. Investment in collected capital, **S**, is defined as:

(2) dS / dt = S' , and

(3) S' = S (V, U, S, D)

where **D** is the stock of collected data called the collected data capital, and **V** and **U** indicate the degree of utilization of **S** and **D**, respectively. Investment in collected capital, **D**, is defined as:

(4) dD / dt = D '

It is done by collecting individual data about statistical units.

To each of the three above processes, **I**, **S**, and **D**, as well as to the storage of **S** and **D**, costs are associated. The cost per time unit are expressed by the cost function

(5) C = C (I,S, D, S, D)

All variables in the system (l)-(5) are time functions.

The objective of a statistical system may be to plan and implement an optimum programme of statistical activities for a period of **T** time units. If the programmes are characterized by the profiles of the time functions **I** and **C** in the time interval of **0 = t < T**, and the evaluation of different pairs of profiles is done by a functional **W**, the problem of optimization may be expressed as finding those time functions of **D**, **V**, **U**, and **M** which subject to the initial values of **S** and **D**, and the conditions (l)-(5),

maximize the functional:

(6) $W = W(f(I)_0^T, g(C)_0^T)$

which is a typical problem of dynamic optimization.

In more elaborated system models, different types of information-supply with unequal value, different types and age classes of **S** and **D** with varying productivities, as well as different types of computations and of collected data with different computation and collection costs have to be distinguished.

The difference between the system outlined above and the traditional system is that the latter do not recognize the data capital, i.e. the computed and the collected capital, as an important factor. The traditional system may be described using the same variables as:

(1') $I = I'(M, S')$

(3') $S' = S'(U, D')$

(5') $C = C'(I, S', D')$

This system says that the statistical information supplied is determined by the multiplication of the current rate of computation of statistics. The computation of statistics is itself determined only by the utilization of the currently collected data. This system has no dynamic elements and therefore leads to maximization of a function:

(**6')** $W = W'(I, C)$

subject to (1'), (3') and (5') which gives a stationary optimum.

In other words, in the statistical file system taking into account data capital, any point of the programme must be a function of the previous time profiles of the variables, while in the traditional system, disregarding that data capital is reuseable, leads to the conclusion that any point or period must be programmed independently of previously computed statistics and collected data.

2.2 Data capital

2.2.1 Data box

The data capital is used as the common name for the computed and collected capital, and it plays an important part in the statistical file system. The condition is, however, that it represents data which are organized in a manner which satisfy the needs of the system. This condition may be illustrated by a data-box containing a number of small rooms for storing data. Each data is identified by the statistical unit to which it is associated and which has its permanent position along the first axis of the box, by the characteristics observed or computed which has its permanent position along the second axis, and finally by the period or point of time which has a permanent position along the third axis of the box. (See **Figure 2**.)

The content of the rooms in a slice of the box across the time axis will give a data picture of the situation at a point or in a period of time. A slice across the axis of characteristics will represent a certain aspect of development, while a slice across the axis of units will tell the registered life story of a unit. The data-box organization requires therefore that we have a system of permanent unit identifiers and standard codes for all characteristics.

By units we mean objects or groups of objects for which there is an interest in finding explanations for typical behaviour, manner of operation or for their general state or change, which justifies that they are distinguished and identified particularly. A unit may therefore be a person, a company, a building, a class of persons, buildings, commodities etc.

A unit which is directly observed may be called an observed unit while a unit which is defined as a group of other units and measured by aggregating values from these units may be called a computed unit. An establishment is usually an observed unit while the chemical industry is a computed unit. Some units may be observed units with respect to certain characteristics and computed with respect to others. A commune or local government district is an observed unit with respect to the items of its budget or accounts, but a computed unit with respect to the gross product of the economic activities within its area. All observed and computed units need their particular positions along the unit axis.

Each characteristic is a timeless class of descriptions. The specific member of each class

corresponding to an observation or a computation is determined by both the time axis and the content of the corresponding room in the data-box. To cover all possible classes of descriptions, we shall need a very large number of positions along the axis of characteristics and we shall have to anticipate future situations to be observed and computed. Usually very few of the total set of possible classes of descriptions apply to each individual unit. The content of the data-box will therefore be very scattered.

These facts make it necessary to investigate the problem of describing an observation or a computed result more thoroughly in order to see if we can find an appropriate solution.

2.2.2 Data description language

The most precise and complete preservation of an observation or a computed result will in most cases require the use of our natural languages. In the data capital this would imply the storage of verbal descriptions which would be expensive to store and process, storage of unnecessary details, and a data capital from which efficient retrieval would be a difficult problem. A constructed language both simple enough to be practical and complete enough to express the main facts precisely may be what we need.

The data-box may be considered as a first step in that direction. The axis of units represents all admissible subjects while the axes of characteristics and times, and the content of the rooms represent the predicates which express the facts about the subjects. An example will indicate the necessity of developing such a language in more detail. Consider an observation of an establishment, **a**, which performs the action, **s**, of selling **x** pieces of commodity, **c**, to a unit, **b**, during a period, **t**. A complete preservation of this observation will require the storage of (**a, s, c, b, t, x**). If the action **s** was unspecified, it might as well have been a description of a purchase and if the purchaser **b** is left unspecified, we have lost a very important relation. In order to give the content of the parenthesis a unique meaning we shall, in addition to defining the meaning of each symbol, have to establish a rule of sequence order to avoid, for example, confusing the seller and the purchaser. Previously the description problems have been solved *ad hoc* in connection with each particular survey by codes and position in cards, etc. When storing and linking data from different sources and surveys, we need more standardized rules for describing facts and we shall call these rules for a data description language [**27**].

The data language consists of a restricted vocabulary of words. The class of the word is determined by a class indicator which is the first letter of any word. We need the following class of indicators:

N: indicates a common noun such as person, establishment, industry, commodity etc.

R: indicates proper nouns of registered units, such as names, identification numbers etc.

V: indicates a verb and represents an action, an event or a state of being, such as produce, sell, wedding, death, be, etc. Only the basic form is needed.

Q: indicates a number such as 300, 0.5 etc.

M: indicates a specification or modification of common nouns, verbs and numbers. The modifiers include such words as "less than", "equal to", "higher than", etc. Measurement units are other examples of modifiers which are used to modify numbers.

P: indicates prepositions which are used to relate units to each other, and are words such as to, from, at, during, etc.

C: indicates conjunctions which are used to join words or groups of words.

The vocabulary of common names includes all the seta of objects we want to observe and analyze. Since we are considering sets of objects, the list of common names represents a listing of the different objects which are surrounding us.

For each set of actions, events and states of being, we need a list of verbs representing each a general characteristic which can be connected to units. The number of verbs should not be very extensive.

The individual units of a set to which a noun refers may be recognized and individually identified in the

vocabulary. A register of units, a classification system for industries, a commodity classification, an industrial classification, a code list for geographic areas, etc., are all examples of parts of the vocabulary used to specify a noun. The same is also true for verbs. The commodity classification, the industry classification, the classification of professions may all be used to modify verbs.

The lists of prepositions and conjunctions are obvious and need no further comments.

A description of an observation is expressed by a sentence of words. The syntax of the sentences can be set out in a few rules:

1. A sentence contains a subject and a predicate,
2. The subject defines the unit described and consists of a common noun and a unit name.
3. The predicate gives the description of the unit, and must always contain a verb. It may also comprise a direct object and/or an indirect object.
4. The direct object describes the possible result of an action or an event and consists of a noun and its modifier.
5. The indirect object describes the unit and relationship to which the subject may be related. It consists of a preposition, a noun and modifiers. Time will always be present in any sentence as an indirect object.
6. Modifiers must always be preceded by the more general words they modify. A modifier may be used in chains for modifying another more general modifier.
7. Equal parts of a sentence may be connected by a conjunction and should then be surrounded by a set of parenthesis.

We shall illustrate the use of the data description language by some examples and choose as the first the above-mentioned observation of establishment a which sells x pieces of commodity c to another unit b during year t. In the description language the sentence describing this will be:

Nestablishment Ra Vsell Ncommodity Mpiece Qx Pto Nestablishment Rb Pduring Ntime Myear Qt

Consider a report about person **i** who moves from district **a** to district **b** at **December 31, 1966**. This observation may be described by:

Nperson Hi Vmove Pfrom Narea Ra Pto Narea Rb Pat Ntime Ndate Q19661231

We may also have computed units which we want to describe. The fact that district **a** at date **t** counted **746** men and **801** women may be expressed by:

Narea Ra Vinclude Npersons (Mmale Q746 Cand Mfemale Q801) Pat Ntime Mdate Qt

Industry a had a value added which was computed to $xx for 1966 and we express this by:

Nindustry Ra Vproduce Nvalue added Mdollars Qxx Pin Ntime Myear Q1966

Frequently we make the same type of observations and perform the same type of computations for a set of units. Instead of repeating the complete sentence for each we introduce the repetition sign, **"**, which means that the class indicator or the word root is repeated. Let us assume that the above computation of value added is done for industries, **a**, **b**, and **c** with the results **xx**, **yy** and **zz** respectively.

We may now write:

Nindustry Ra Vproduce Nvalue added Mji Qxx Pin Ntime Myear Q1966

" b " " " " " "yy " " " "

" c " " " " " "zz " " " "

We proceed still a step further toward our aim by introducing the dummy substitute sign, **-** , which makes it possible to distinguish the common elements of the sentence from the particular in a set description sentence followed by the particularities of each observation:

Nindustry R- Vproduce Nvalue added M$ Q- Pin Ntime Myear Q1966

“ “a “ “ “ “ “xx “ “ “ “ ”

“ “b “ “ “ “ “yy “ “ “ “ ”

“ “c “ “ “ “ “zz “ “ “ “ ”

The latter form has an obvious resemblance to the traditional way of working in which the set description is verbally formed either in a procedure description or perhaps printed on the cards while only the content of the three last lines are punched in cards and accessible by machines.

The special status of a unit expressed by the verb **be** is described as modifications to the register unit, e.g.:

Nperson Rxxx Mfemale Vis Pat Ntime Mdate Qt

It should only be used if the status cannot be described by another ver

By establishing a controlled limited vocabulary of words and a language syntax we may achieve three important things. One, we are able to describe complicated facts in a compact and concise way. Two, we have a standardized way of describing observations which will be of great value for efficient communication between data collector and data user through the data capital. Three, we have a description which can be handled by automatic data processing equipment.

Let us return to the data box representation of the data capital. We can now state that the subject specifies the position along the unit axis, the indirect object of the time represents the time axis specification, the value or quantity is the number we want to put into the data room, while the rest of the sentence express the specification along the axis of characteristics. To draw up this last specification directly anticipating all possible types of observations would probably be an impossible task.

The language outlined above is a description language for the data capital. To describe the information system completely, a description of collecting, computing and publishing procedures would be needed. Such a procedure description language could be constructed according to similar lines as the above and be used for storing and retrieval of all procedure details. This aspect is, however, considered to be outside the frame of the present paper.

The outline description language is only one, and may be not the best approach, to preserve as many as possible of the characteristics of observations or computed data. An alternative approach may be to use a standard form with a fixed number of fields for description. This means, however, in terms of the above discussion that a fixed sentence structure is always used. As a minimum number of words this standard sentence structure must at least include fields for acting unit, receiving unit, action, object, two time fields defining start and finish of time period and a field for measurement value. Such a standard sentence structure will, as the above example may indicate, in some cases be unnecessarily voluminous, while in other situations it will not be able to take care of all necessary aspects.

Finally, it should be emphasized that this section should not be considered as a comprehensive discussion of the problem of compact and concise description of statistical facts, but rather as a proposal for further discussion.

2.2.3 Table description language

The above description language may as well be used for description of computed as for collected data capital, and should indeed be. However, there are certain sets of computed capital components which are stored only for the purpose of being reproduced exactly in the form and configuration in which they originally were produced. Examples may be certain statistical tables which are stored only for the purpose of being completely reproduced [**23**].

In these cases, the most convenient description of the data may be a description of the tables by means of their names, heading and front column. This type of data description is called table description.

It may seem that this is a more precise and useful way of description than the description language, but it is important to notice that already at the present stage the possibilities for automatic handling and linkage are very limited for this form of description.

We shall assume that a table description consists of a unique table no., in addition to the name, heading and front column texts and has the following form:

TABLE a text COL text, text, text, ... ROW text, ... text

which may be called the table description language. As we can see, this is a fixed format language.

3. Data files

3.1 Automatic Data files

3.1.1 Construction

The real counterpart to the theoretical data capital is the data files. In this part of the paper, we shall discuss problems in connection with the establishment and maintenance of data files for statistical purposes.

There are two types of problems which have to be faced in this connection.

These are the problems of the logical construction and organization of the files which are common for, but independent of, any particular data processing system, and the problems of the physical implementation of the logical file system for a particular data processing system. In order to avoid discussion about a particular data processing system, this paper will be limited to the first group of problems [**3, 5, 6**].

An automatic data file is based on logical sets organized according to certain classification rules. Each set has a name. There are two types of sets which will here be called reference sets and data sets. The reference sets determine the file structure and organization.

Each set consists of one or more logical records. The record may be further subdivided into fields which primarily contain a word, a quantity or a name. The reference set is characterized by containing records which each consists of one logical entry and one exit part. The data set on the other hand has records which may have entries, but which have no exit parts. The data sets contain the data on which the computations are performed and may therefore be called the terminal sets within the file system. (See **Figure 3**)

All sets may be considered as organized in a hierarchy with a reference set as initial set, and all other sets are either directly, or through other reference sets, subordinated to the initial set. A set or a record may therefore be named by the chain of set names from the initial set to the terminal set.

3.1.2 Storage devices and media

An automatic file is kept on storage media which are handled by storage devices incorporated in a data processing system. Punched cards and magnetic tapes are examples of storage media, while card units and

tape stations in this context are storage devices. The storage medium and device may be inseparable which is the case for core stores, magnetic drum stores, etc.

The storage medium may either be an on-line medium or an off-line medium depending on whether it is mounted on a device and ready for automatic handling or whether it is demounted and requires manual adjustment to be accessible for the system.

The storage devices may be classified either as a direct access or a sequential access device. The degree of access is expressed by the time required to make a specified position on a specified medium available to the file system. A direct access device has equal access time to any position on the medium it handles. In a sequential access device the positions on the storage medium are available sequentially according to a predetermined ordering and the access time to a storage position is proportional to the distance between this position and to the position which at the moment is available.

Even though most data processing system configurations will include both direct and sequential storage devices, it seems reasonable to assume that the sequential storage type of device will play an important part in our type of file system because of its relative good capacity/cost ratio.

The extent of the data files will also make it necessary to assume that large parts of the files must be

kept on off-line media.

3.2 Reference sets

3.2.1 Vocabulary sets

An important class of reference sets are those containing the vocabulary of the description language. The entry and exit fields in each of the records of these sets contain the external and the internal dialect word, respectively. The external word refers to the language used outside the file system while the second field gives the words of the dialect used by the filing system. There will also be a need for vocabulary sets in which the contents of entries and exits are exchanged.

We distinguish between two classes of reference sets both belonging to the vocabulary. They are the unit registers and the description code lists.

3.2.1.1 Registers

A register is here defined as a list of all units of a certain type. It may be appropriate to subdivide a register into several sets, e.g. by the geographical location, the age, etc. of the units.

The external name of a register unit is the name by which data outside the system may be matched to the registered statistical units. The most common name is the natural name and address of the unit. However, the name and address may be ambiguous and it may change over time, and a risk of erroneous matching exists. The name field may therefore be extended to include other data about the unit, e.g. birth date, to reduce the risk for mis-matching. The condition is of course that these additional data are requested together with the name from respondents. It is therefore important to establish names and matching procedures which minimize the risk for erroneous matching of units. [**14, 15**]

In many statistical applications we may name the questionnaires before distributing them to respondents. In this situation we may extend the name to include a serial identification number in addition to the natural name. The returned data can then easily be matched with the register, and the risk for errors may seem eliminated. The fact is, however, that the mailman or the interviewer has been charged with the matching responsibility, and the risk will still be present and perhaps may not be subject to control.

The ideal situation is when we are able to supply each unit with its serial number and get the respondents themselves to identify their reports with the assigned unambiguous serial numbers. For several reasons this situation will hardly ever occur. The reasons are that people are reluctant to accept and use the serial identification numbers, that the definition of the units to which the numbers are assigned may be misunderstood, that the numbers for several units reported for by the same respondent are confused, that the numbers are influenced by reporting and processing errors, etc.

In order to eliminate some of the sources of error in such serial numbers, they may be extended to include one or more check digits. A check digit may be regarded as the value of a function, the argument of which is the serial number. The identification number to be used will then be I.he serial number, with the check digits written, for example, to the right of the least significant digit of the serial number. By a re-computation of the function value a check on the validity of reported identification numbers may be performed.

A common check digit generation procedure is the modulo N cheek, the principle of which is that the serial number is divided by W and the remainder used as check digits. This basic method may be modified by, for example, weighting each individual digit of the serial number. In this say the error detection effect may be maximized given the probabilities for different error pattern types. The check digit computation may also be repeated several times each time including the previously computed check digits in the dividend.

Some work has also been done to develop self-correcting numbers [**2, 4**]. When an error is detected in the control computation, an automatic correction of the number is carried out in such a way that it will pass the control. This does not, however, guarantee that a correct match is obtained. We shall always have to consider a certain risk of mis-matching.

At this stage it should be emphasized that the identification numbers are supplements to, or substi-

tutes for, the natural names and are introduced to reduce the risk of erroneous matching. The check digits are of little use when the identification numbers, for example, are pre-printing on documents which are later automatically read. It is also unnecessary and waste of storage space to store the check digits on the medium which holds the register since the processing system will usually be able to compute them during printing without any additional time consumption. When collected and checked the check digits are of no further value and should be discarded before storage of the data.

The internal name field of a register record is used to hold the identifier used within the automatic file system. The internal identifier may or may not be identical to the external identifier. We shall return to the characteristics of the internal identifiers in a later section.

The number and type of registers needed will of course depend on the local conditions, the stage of development of the statistical system, etc. We may, however, try to establish a tentative list of registers which may comprise registers of:

a. Persons
b. Companies (non—physical, juridical bodies)
c. Enterprises (including governmental enterprises, associations)
d. Establishments
e. Land properties
f. Geographic, administrative areas (including foreign countries)
g. Commodities (including livestock, buildings, financial assets)
h. Industries
i. Households

These registers are all included in the **R-set** of the file.

3.2.1.2 Code lists

A code list is a set of words within the description language which are not names of statistical units. There is in fact very little difference between a register and a code list, and most of what has been said above about unit registers will also apply to code lists. The collection of code lists must cover all relevant objects other than units, activities, events, states of being, etc.

A central problem in the preceding section was the matching of units for which data were collected and units in the registers. A similar matching problem exists also in connection with the code lists. We want to reduce the risk of classifying a part of a sentence wrongly. This risk occurs because different people may use the same notion for unequal meanings or concepts, because errors may be introduced by a confusion of questions when reporting, etc.This problem has been approached by construction and use of editing procedures, the aim of which is to detect errors and avoid mis-matching. The editing methods may therefore in this context be regarded as methods to minimize the risk for mis-matching.

The set of code lists seems to have to include the following lists:

a. List of other nouns including the names of registers, the N-set.
b. List of prepositions, the P-set.
c. List of conjunctions, the C-set.
d. List of selected verbs, the V-set.
e. List of other words in vocabulary, the M-set.

The list of selected verbs should be a rather short list of general verbs indicating the main classes of action. It should be restricted to such a level that even for wide classes of actions only one verb would be relevant. The same verb should, for example, cover the class including to bear, to create, to produce, to make, while the verbs to die, to cease, to finish, etc. should be covered by the same general word. The specification of the action is obtained by the words in list e when used as modifiers to the verb.

3.2.1.3 Internal dialect

The data description language may be considered as having two dialects both with common syntax. The external dialect is used in communication between users of the file system and the file system, the second only by the internal file system processes. The internal dialect is represented with the words in the second field of the register and code list records.

Space requirements and operation times are essential factors in the file system efficiency and the construction of the internal words should therefore be dictated by these factors. In general, this implies that the internal words should be made compact with as small a redundancy as possible. Assuming that the technical reliability has now reached a very high level, we may eliminate all kinds of check-digits etc. from the internal words. This may result in a mechanical assignment of different character combinations to the words from a list of combinations of the different characters known to processing system by first utilizing all single characters, then all combinations of any two characters, all combinations of any three characters, etc. We may now find it efficient to assing the combinations of few characters to words which appear frequently while combinations of many characters for words appearing less frequently.

The internal dialect is to a large extent dependent on the particular data processing system by which the file system is implemented, and we shall therefore leave the discussion of it at this stage.

3.2.2 Catalogues

In addition to the sets which comprise the vocabulary of the description language, we need other reference sets which give the necessary references between sets, either directly or indirectly to a data set. We shall call sets which give references to other sets catalogues. The entry fields in the catalogues contain external names while the exit contain real set names or labels.

3.2.2.1 Master set

The sets may be considered as a hierarchy of which the highest set is the initial master set. The master set is simply an inventory list of all second level sets in the system. Any use of the file system implies the participation of the master set.

The entry fields of the master set contain the external names of all sets referred to by the master set, while the corresponding exit fields give the real names of these sets. The real names may be exactly the same as the external names in which case the master set only plays the role of an inventory list. On the other hand, when external and real names differ the master set plays a role similar to the vocabulary with the difference that the master set refers to sets and not to words.

The real set names will usually be meaningless outside the system.

3.2.2.2 Security sets

The data files may contain data which should not be used by everyone having access to the file system. The section in charge of the information service should undoubtedly have access to the file, but not to those parts which contain confidential data on units. Subject specialists working in one field should not have access to other fields without authorization. Analysts should not be permitted to obtain any kind of data unrestricted.

The section of the statistical system which operates the file system will be a technical operating staff which should not be charged with the responsibility to supervise each use of the filed data. As far as possible, this supervision should be automated to obtain the required data security.

This may partly be implemented by the security sets. A security set is a set the name of which and/or the individual entries represent secret passwords. The corresponding exit fields contain the keys which are real names of other sets. To obtain the key which unlocks a set, the user must supply the correct password.

There may be several levels of security. The first password may, for example, refer to an entry in the master file. The corresponding key is the name of a security set in which the second password is an entry and the key of which may be the name of a second security set and so on. The passwords may be supplied separately by different persons, and the use of a particular data set may therefore be made dependent of several persons' authorization.

We should, however, never forget that as a locksmith can manage to unlock a safe, so can a system programmer unlock any data set if he gets free access to the system.

3.2.2.3 Sets of descriptions

A very important class of reference sets is the sets of descriptions. The real name of such a set •ay either be an exit of the master set in which case this is a set of open descriptions or it may be an ea of a security set in which case it is a set of locked descriptions.

Each entry field of a set of descriptions contains a data description formulated in the data or table description languages. A list of all entry fields of all description sets would therefore give a comprehensive description of all data sets stored in the file system. The exit fields of the sets of descriptions contain usually the real names of the data sets corresponding to the descriptions. Alternatively, the exits may refer to 'the real names of security sets in which case the descriptions may be generally available while the data sets are locked.

3.3 Data sets

According to the approach followed above it will be natural to distinguish between unit data sets and table data sets, which both may be considered as terminal sets.

3.3.1 Unit data sets

Unit data sets are those sets containing records described by data description sentences and referred to by the description sets. There may be a varying number of records within the data sets. In certain applications the individual record is considered as a set itself the name of which is one of its fields.

The unit data sets contain the data which really are the most valuable part of the file and which may be linked and utilized in further computations.

3.3.2 Table data sets

Table data sets each contain the matrix of a previously prepared table which is described in a table description. These sets together with their descriptions cannot in general be utilized automatically in further computations but are mainly intended for later reproduction in their original form, i.e. in new issues of automatically prepared publications, or as displays on inquiry devices connected to the file system.

3.4. Summary of the file structure

It is pointed out above that the file structure may be considered as composed of a hierarchy of sets at different levels. This may be illustrated by the following table:

<table>
<tr><td>Level 1
Level 2
Level 3
Level 4</td><td>Security status</td><td>Security status</td></tr>
<tr><td rowspan="2">Master set: M
Security set: S
Data description sets: S.P1
Unit data sets: S.P1.UD
Data description sets: D
Security set: D0UD
Unit data sets: D2.UD.P2</td><td rowspan="2">Locket data set</td><td>Locked description sets</td></tr>
<tr><td>Open description sets</td></tr>
</table>

Data description sets: D Unit data sets: D.UD Table description sets: T Table data sets: T.TD	Open data set	Open unit data sets Open table sets
Vocabulary sets R/H/V/P/	Open vocabulary sets	

The above table indicates one out of many possible structures. In the cases of locked data sets, the approach to the terminal sets goes through level 1 to level 4, while the open vocabulary sets may be reached by a reference from the master set.

A. hierarchy of this type is often compared to a tree, the root of which is the master set and ends of the branches the terminal set. The identity of a set may be given as the branch from which it is derived. We may by using the master set as an initial point, identify any set by considering the entry names of the master set as the names of the second level sets, e.g. naming the security set S, and considering S combined by any one of the security set entries as names of the third level sets branching from the security set, etc.

An nth level name will in general consist of the combination of (**n-1**) entries needed to reach the set. We shall call such a name a symbolic set name. The symbolic set name must always be used in communication with the system because the real set name is only known internally to the file system and is identical with the content of the exit field corresponding to the last entry. The symbolic name **S.P1.UD** where **P1** is a password and **UD** is a unit data description is the symbolic name of a data set. Its real name is only specified in the exit field corresponding to the entry with the content **UD** in the description set **S.P1**. This implies that a data set cannot without much effort be identified without knowing its complete symbolic name. Access to the physical storage media does not, therefore, imply possibility of misuse, (see figure 4).

In the following discussions, we shall use the above outlined file structure as an illustration. That means we shall consider table data and vocabulary lists as open sets. Data descriptions or unit data sets may be both open and closed. This is of course only one out of many possible structures.

4. File processes

4.1 General considerations

We have now established the set components of the file and are to embark on the discussion of the processes of the file system. We shall start with the observation of the work done by librarians through centuries which more recently has been adjusted to the possibilities offered by modern data processingequipment. In a library, books etc. are received, classified, catalogued and stored away on shelves. This may be called the storing mode of the library function. The second mode of the library function is the retrieval of books. A library user has a certain problem or field of interest and wants to obtain those books which deal with the subject in question. He starts with the catalogue and looks up those key words he thinks cover his field. He will usually find a number of references and by means of further specifications such as author, book title, etc., he selects the references to the books he wishes to see. With the help of these references the books are finally retrieved. A third mode is the deletion from the catalogue and the exclusion of books from the shelves when, for example, scarcity of space may require that infrequently used books are taken out of the library to make space available for new books which are in greater demand. [**1, 27**]

There are many similarities between the work described above and the data file work and it may be useful to consider what has been done by librarians. Much work has, for example, been done by the librarians to take advantage of modern data processing equipment, and in this connection special index systems for efficient classification have been developed, description languages proposed, search techniques investigated, implemented and improved, as well as special complete machine oriented systems, which may be worth while studying.

In the present paper we shall concentrate on the logical aspects and start by listing the different tasks which we associate with a file system:

a. Describing data sets
b. Ordering data sets
c. Translating descriptions to internal dialect
d. Storing descriptions and data sets
e. Specifying inquiries for data sets
f. Translating specifications to internal dialect
g. Retrieving descriptions and data sets
h. Retranslating internal descriptions
i. Specifying deletions of descriptions and data sets
j. Deleting descriptions and data sets

Of these ten tasks, **a., e**. and **i**. must be done outside the file system. Task **b**. may be considered as an auxilliary type which represents sorting, merging, and matching of data sets. This is a task for which there exist well developed and known solutions which should need no further comments in this context.

The file system may be considered as working in one out of three working modes, the storing mode, the retrieval mode or the deletion mode. We shall call the working rules which the system follows procedures. and investigate which procedures are necessary to perform the task listed as **c**., **d**., **f**., **g**., **h**., and **j**. above. (See **Figure 5**)

4.2 Description of inquiries

The task of describing the mass of information about a data set in one or a few description sentences will probably still for some time have to be done manually by specialists in order to get uniform descriptions. The aim is to get to a state in which any data set described by any two describers results in two equal description sentences. The vocabulary without synonyms and the simple sentence syntax will to a large extent contribute to the establishment of such a state. However, there will still be room for different interpretation and evaluation of the available information about the data set and two persons may have different opinions. By reducing the number of persons describing data sets, they will obtain more experience and learn from each other, which will provide more conformity in descriptions. Ultimately, this process also should of course be automated.

The description of inquiries is another task which is quite similar to description of a set to be stored. The problem here is to describe the data set which is wanted. The description should either be made in terms of the data description language, in case the required data set is supposed to be a unit data set, or in terms of a table description, in case the required set is a table set. A few examples will illustrate the description task.

We want to retrieve the data set containing information about all establishments which were producing something in 1965. The description of this in a sentence would be:

Nestablish R- Vproduce Pin Ntime Myear Q 1965

The word **R-** indicates that the names of the establishments are without any significance in the definition of the set, but we want to have the data specified by unit identification.

Let us now suppose that we have retrieved this set from the file, but we do also want the set of persons employed by these establishments at **December 31, 1965**. This record inquiry may be described as:

Npersons R- Vwork Pin Nestablishment R- Pat Ntime Mdate Q1965 1231
“ “ “ “ “ “xx “ “ “ “
“ “ “ “ “ “yy “ “ “ “

.

where the list **xx**, **yy**, etc. indicates the units obtained by the first stage request.

A specification may also make use of words like **greater than**, **equa**l, **less than**, etc. We may want the set of persons who were born in **1920** or later, and may specify this by:

Npersons R- Vborn Pat Ntime Mdate (M equal C or M bigger than) Q 1920 0101

The above example illustrates also the use of the conjunction **or** which is equivalent to a logical **OR** operator.

We may also describe sets which are defined by the logical **AND**, for example, if we want the enterprises which belonged to a certain area **a** and were employing more than **100** employees at **September 1, 1963**.

Nenterprises R- (Vincluded Pin Narea Ra C and V employ N persons M bigger than Q 100) P at Ntime Mdate Q 1963 09013

A description must cover two purposes. First, it should give the criteria for selecting the units which should be included in the set, and second, it should specify the data we want about these units. We try to formulate the description in such a manner that we may obtain the maximum amount of relevant data without excluding a given upper limit for including irrelevant information.

In the same way as the description of a given data set, the description of a wanted data set will probably have to be done for some time by experienced people, even though the aim must be automatic description formulation from a free verbal form.

Inquiries for table sets may be described by similar rules. If the table number is known to be 351, the necessary description will be

TABLE 351

On the other hand, if we want all tables referring to 1966 we may form the description

NAME - 1966 –

which will indicate that all tables, the names of which include **1966**, are wanted.

This may be extended to specify that tables including 1**966** in name, text column or row are wanted:

NAME - 1966 - , C or, COL - 1966 - , C or, ROW - 1966 –

4.3 Working modes

The working mode and the working specification is given by mode name and symbolic set name, and we distinguish among three mode names **Store**, **Retrieve** and **Delete**. The mode name may be considered as the name of a set of instructions necessary for working in the respective modes. The set name has already been discussed and may be of the form **S.P1.UD**, **D.UD.P2**, **D.TJD** and **T.TD** assuming the model structure of section 3.A. **UD** and **TD** may be descriptions of sets to be stored or sets to be retrieved or deleted. The general form of a process specification has the form:

Mode name: Symbolic set name

4..3.1 Storing mode

The process specification for storing may be of the form:

Store: S. P1. UD

which means that a unit data set with the symbolic name **S.P1.UD** should be stored.

The work proceeds in the following manner: The master set is searched for the record with the name **S** as an entry. The exit of this record supplies the real name or reference to the security set. Then the security set is searched for a record with an entry corresponding to the password **P1**. Now two situations may occur, the password may or may not exist. Let us first take the latter case. Then a new record is established in the security set, the entry field of which is the password **P1** and the exit field a

real name assigned automatically by the system. A new description set with this real name is established. The description **UD** may now need translation to internal and more compact representation. This is carried out word by word of **UD** starting with the word class indicator. The master set is searched for an entry equal to the indicator and the corresponding exit gives the reference to the relevant code list or register. The code list or register is then searched for the entry corresponding to the word to be translated, and the exit gives the internal word wanted. This is done for all words of the description.

When the description is translated we establish one record in the new description set and write the translated description in its entry field. The system generates a real name for the data set which is both written in the exit field of the description record and used as a real name of the data set which is read and translated.

When the password already exists in **S**, it may be because the description of the present set really is wanted stored together with other secret descriptions or it may be a functional relationship. Neither case seems to represent any problem and both can be treated in the same manner. No new record is needed in **S** and the already existing description set **S.P1** is used for storing the present description **UD** in a new description record after being translated. The data set is then read, translated and assigned the real name generated by the system.

Storing table descriptions are extremely simple and the specification may be of the following form:

Store: T.TD

The master set is searched for entry **T** and the real name of the table description set obtained. This set is searched for an entry **TD**. Normally the **TD** description does not exist. A new description record is established, with **TD** in the entry field and a generated real set name in the exit field. The set is then read and identified by the generated real name.

4.3.2 Retrieval mode

When working in the retrieval mode the task is to find one or more possible data sets with a specified symbolic name. Four situations may occur. First, no set exists with the specified name. In this case, we may want to make the specification wider. Second, one stored set exists with the specified name. Third, several stored sets exist. This may be considered as a generalization of the second situation. The division into store sets is often arbitrary from a retrieval point of view. When the response to an inquiry is several stored sets, this may be regarded as one logical set from retrieval point with the specified name. The reason for the appearance of this situation is that the description may use dummy symbols, logical operators, etc. to define a new set description which refers not only to one, but several stored data sets. Fourth, we may have the opposite case in which we only want selected records of a data set. The typical retrieval specification is:

Retrieve: D.UD

where **UD** is an inquiry description.

By the master set the reference to data description set **D** is obtained. The description **UD** is translated and the description set searched. If no match is obtained, the retrieval is terminated. On the other hand, if one or more matches are obtained, the corresponding data sets are copied, edited and translated into external language, if necessary.

The inquirer may have authorized access also to locked descriptions and will then specify the retrieval in one of the following forms:

Retrieve: S. P1. UD

Retrieve: D. UD. P2

assuming he possesses the passwords **P1** or **P2**. Retrieval of table data sets is straightforward along the lines outlined above.

4.3.3 Deletion mode

With restricted storage space or capacity, deletion of stored data may be necessary. This process is quite similar to the retrieval with the difference that a stored description which is matched with the specified is deleted and the corresponding data set is also deleted instead of copied.

There is, however, a rather important aspect which should be mentioned in this connexion. Very valuable information may "be destroyed if somebody starts deleting data sets without authorization. In the same way as the data sets may be locked, we mdy introduce locked sets for working instructions. These sets are identified by their names, i.e. **Store**, **Retrieval**, **Delete**. One simple way of solving the problem of avoiding unauthorized deletion is by giving this mode a secret name. Delete might therefore be considered as a stand-in for a secret mode name.

With the above modifications, the deletion mode should be obvious and initiated by a process specification, e.g.

Delete: D. UD

with the effect of deleting both the data set and the record of **D** containing the description **UD**.

4.4 System procedures

So far, we have discussed the file operations in a rather general manner. In the following sections, we shall proceed by considering the operations in more detail, but still avoiding as far as possible touching the pure machine related problems [**17**].

Just as the file structure was described in terms of sets, so we shall consider the file operations in terms of procedures which in fact are sets containing instructions. A procedure is a set of working instructions which together with the available data processing system unambiguously determines how a defined task should be solved,

A procedure may read, process and write on-line sets. The sets handled are either input sets or output sets to the procedures. An input set may be a specification, or a set to be stored, retrieved or deleted, while an output set is a written message, or a set written for storage, or a set retrieved and copied. The distinction between input and output sets is therefore relative to the specific procedure by which they are handled.

The different working modes partly make use of the same procedures and there is therefore no one-to-one correspondence between procedures and modes.

4.4.1 Master procedure

The master procedure is that procedure which takes care of everything not handled by any other specified procedure. It has two main functions. First, it should be the backbone of the file work which ties the other procedures together in meaningful logical file processes. Second, it has to be the link between the logical file procedures and the machine system routines. It is the first aspect which is discussed here. It will be related to the particular structure of file components used above, but it should be understood that similar master procedures could be developed for any other file structure.

The master procedure recognizes two kinds of input sets, the process specification and the data set, two kinds of output sets, the message and the data set, and seven types of resident reference sets, the **M-set**, the **S-set**, the **D-set**, the **T-set**, the **S.Pi-sets**, the **D.UD-sets** and the **V-set**.

The message is the final result of any file task, and is a texted output which gives one of the following pieces of information:

a. No entry in **M-set** corresponding to specification
b. No entry in **T-set** corresponding to specification
c. No entry in **D-set** corresponding to specification
d. No entry in **S-set** corresponding to specification
e. No entry in **D.UD**-set corresponding to password
f. No entry in **S.P1**-set corresponding to password
g. Get off-line medium with given real name on-line
h. Store on-line medium with given real name off-line
i. Set with given symbolic name is retrieved
j. Set with given symbolic name deleted

k. No entry in W-set for input data set

The **W-set** introduced above is a working reference set in which data sets requested are registered in case they are not on-line. If the mode is the storing mode, the entry field will be the symbolic name of the set and the exit field will be the assigned real name. On the other hand, if the system is working in retrieval or deletion mode the content of the two fields is exchanged. The **V-set** will never be known or accessible outside the master procedure of which it may be considered a part.

We assume that all required reference sets are kept on-line while the data sets may either be online or off-line. This does not imply that there need to be on-line capacity for all reference sets, but that those needed for a particular task are on-line when it is processed.

The master procedure makes use of six special procedures: the **Read procedure**, the **Search procedure**, the **Translate procedure**, the **Update procedure**, the **Delete procedure** and the **Write procedure**. The names of the procedures will give a preliminary idea about their respective functions. The master procedure ties these procedures together into meaningful file processes by stepwise decisions based on the content of the process specification, the result of search and the on-off-line status of the data set. The design of this network of decisions is most appropriately illustrated by a flow diagram. (See **Figure 6**)

A few comments may be useful in connection with the flow diagram. As can be seen, two different main paths may be followed depending on whether the input set is a process specification or a data set. The organizational work is mainly carried out along the process specification path, e.g. the checking of passwords, the translation of descriptions, the search for descriptions, etc.

If, however, only a subset of a previously stored data set is wanted, the search within the data set is carried out in the data set path in the following manner: When a data set is identified, e.g. as **D.UD**, and the description in addition specifies certain records within data set, each of these are considered as an independent set and stored in the **W-set**. Later when the data set **D.UD** is read, each specified record within the set is treated as a data set and checked against **W**.

4.4.2 Read Procedure

The first procedure called by the master procedure will always be the **read procedure**. The task of this procedure is to make an input set available to the **master procedure**.

The read procedure must be able to read two types of input sets, the process specification and the data set. The latter may either be identified by its symbolic or its real name.

It is here assumed that the necessary technical description of the set follow the set and is automatically checked by the computer oriented routines.

4.4.3 Search procedure

The **search procedure** gives the instructions for searching a specified reference set. The necessary specifications for the search procedure will be the file to be searched, and the entry name or names searched. The output of the search procedure will be the content of the exit field or fields of records with the searched entry or a sign indicating a negative result of the search.

There are many possible ways of designing such a search procedure. We shall assume that a sequential search is carried out. A sequential search follows a forward or backward predetermined path through the storage medium keeping the set to be searched. The path along a sequential medium, for example a magnetic tape, is obvious; for direct access media we assume that a path can be determined conventionally.

4.4.4 Translate procedure

The **translate procedure** takes care of the translation of record set descriptions from the external dialect to the internal dialect and vice versa. The input to the translate procedure is a set of words of a description and the output is a similar set of words in the opposite dialect.

The translate procedure assumes the availability of the vocabulary **N-set**, **R-set**, **V-set**, **M-set**, **P-set**, **C-set**, the inverted versions and the master set, the search procedure and the write procedure. The search procedure is needed to search the register and code lists, while the write procedure is needed

for writing messages when invalid words occur.

4.4.5 Updating procedure

The **updating procedure** takes care of creating new reference records or deleting existing records. The first function implies that the updating procedure is responsible for assigning real names to sets. It also makes use of the write, search and delete procedures.

The input to the updating procedure is the mode, the real name of the set to be updated and the symbolic name of the record to be deleted or created.

According to whether the mode is store or not, a real name is generated. In store mode the symbolic name given and the real name generated is written as a record in the set specified by real name. If such a set does not already exist, the record is named as a set with the given real name.

In the delete mode, the set with real name corresponding to the specified is searched for the record with the specified symbolic entry which is then deleted.

4.4.6 Delete procedure

The **delete procedure** is a procedure which deletes the set. It requires the real name of the data set as an input.

4.4.7 Write procedure

The **write procedure** is responsible for copying data sets in the form they are required by the process and for writing standard messages which have already been discussed.

It requires a process specification, a data set and/or the result of a decision.

5. File organization

5.1 Organization objectives

We have discussed the formal structure and operation of a statistical file system. In the following sections we shall discuss the organization of records within a set and the organization of the file into sets [**9, 10, 11, 12, 25**].

The organization objectives may be the design of a file system which minimizes the storage and retrieval time and/or the cost of operation in general. We shall in this context consider the composition and organization of' input and output sets as given and assume that the means for achieving the objectives are the alternative possibilities for partitioning the data file into sets, duplication of data into multiple representation in different sets, allocation of sets among different storage devices, and the ordering of records within each set. It may be useful to keep the picture of the data box in mind when discussing these problems.

Important work has been carried out in developing models for optimum file organization. Each field of application has, however, its special features which may be of importance for the solutions. In statistical file work, it may be important to emphasize the great variation in the output set requirements which makes it necessary to link data from different sets to a large extent. It is also worth noting the central part which the unit plays in a statistical file system, and it is assumed that we shall frequently get inquiries for time series on unit basis.

Another aspect which should also be mentioned in this connection is that we shall not be able to specify the characteristics of the future inquiries with certainty, which is often the case in many other file systems. In the statistical file system we will have to anticipate the output set structures which may influence the file organization.

The ultimate aim must be the development of a system in which the file organization automatically adjusts itself to the changing conditions of input, file and output composition.

5.2 Organization within sets

5.2.1 Ordering of records

The ordering of records is determined by ordering rules which determine the sequence through the storage medium. The ordering rules may either refer to the. record name or to characteristics associated with the processing of the record, and we shall therefore distinguish between name dependent and name independent ordering.

Typical name dependent ordering is ascending or descending ordering by name. The name is usually defined as the content of a special part or field of the record. In reference sets, the content of the entry field will usually be the natural name of the record. A special case is the one in which the record name is kept in the entry field of a reference set, the exit of which is the position of the named record at a storage medium.

In unit data sets there are several possibilities. We may use the subject as a record name. This seems to be a very important situation because we shall easily be able to link together data from different sets for the same subject. But we may also think of other names, for example certain indirect objects appearing in each record, e.g. the time to which the data refer. Frequently, we shall have several records for the same subject and several records referring to the same time, and it will be necessary to establish more specified names for the records. We may name the record with the combination of several parts as the subject and the time specification, etc.

The name independent ordering may be that in which the records are received from the respondents. This type of ordering is also referred to as random which, however, may be a less adequate notation in this context.

We are interested in the ordering of input sets, file sets and output sets, of which we only consider the ordering of the file set within the control of the file system. File maintenance is considered as the processing of an output set which is a copy of the original set followed by the storage of an input set which is the updated set. We assume that any file set is established by reading and storing one input set. Given the ordering of the input and output sets which file ordering will be the most efficient?

5.2.2 Efficient record ordering

The operation time may be affected by the record ordering of a file set, by the time used for re-arranging the input set in the ordering required by the file and by the time required to rearrange the output sets to the ordering required by the users.

The time needed for rearranging the records of a data set will in general increase more than proportionately with the number of records. If this relation is denoted by:

$$T = 0(S)$$

we shall in general have that

$$O(S^I) \geq O(S^U) + O(S^U)$$

where S^I is the input set and S^U and S^U two output sets.

This indicates that it is more efficient or at least as efficient to accept the input set ordering as a file ordering and rearrange the output sets if their total size does not exceed the size of the input set, than rearranging the input set to the expected ordering of the required output sets. Even when the total size of all output sets exceeds the size of the input set, we may frequently find that the above decision may be efficient. An extreme example is the case where all output requests refer to a single record only, in which case any rearrangement of the input set would be wasted.

Let us consider the situation in which we expect R requests about data from the file and in which case we assume that pI is the probability that a request will imply an operation time ti for rearrangement of the output set subject to a file based on the original input order. The change in operation time because of a rearrangement of the input set will then be:

$$\Delta T = 0(S) - \Sigma(p_i - p_i') \cdot t_i \cdot R$$

where p_i' is the probability that an output set will require a rearrangement time t_i subject to the new file order. Whether the rearrangement of the input set will be efficient or not, will obviously depend on the number of expected requests and on the possible reduction in the average expected rearrangement time per requested output set implied by the rearrangement of the input set.

Frequently, a file set S is sooner or later copied completely to be the basis for a new set updated by new data. If the input set is rearranged in the ordering required for the generation of the new set, the change in operation time will be:

$$\Delta T = - \Sigma(p_i - p_i') \cdot t_i \cdot (R - 1)$$

If this ordering also reduces the expected time needed for rearrangement of the other **R-1** requested sets, the rearrangement of the input set is efficient. Which is the most convenient ordering for up-dating a set by new data?

The first criterion is probably that the relative ordering is invariant. Such an ordering is obtained if based on a permanent record name, that is a name derived from characteristics for the individual unit which do not vary. If ordering is alphabetical, based on the names of, for example, persons observed, the names may change because of marriage, etc. When this occurs, we shall have to make updating of two records, i.e. deleting one based on the relative position determined by the old name and creating a new record in the relative position determined "by the new name which, of course, could have been avoided if the records were ordered according to some non-variable characteristics. A second criterion for the updating order is that the name must be unambiguous. It is therefore not usually sufficient to name each record by birth day of the unit to which it refers, because we may then again encounter the mis-matching problem discussed above.

To conclude, an ordering based on a permanent unit identification, such as the permanent identification number, seems to be an ideal ordering for the file set subject to the updating require-ment.

Even though we know that this arrangement of records will in general be the most efficient ordering independent of the requirements of other output sets, it may be important to investigate whether this ordering deviates essentially from those required by users of file data.

As has been emphasized in several places above, one of the most important gains from the file system will probably be the possibility of linking data for identical units from different file sets representing different subject fields, points of time or time periods. The natural link will again be the permanent identifications of the units, and the requirements of the output sets' ordering will therefore coincide with that of the updating, i.e. both will require an output record ordering by permanent unit identification.

We may therefore conclude this discussion by stating that an ordering based on the permanent unit identification also seems to be the optimum file ordering and that any input set not in this ordering should be rearranged.

In this connection it should be noted that we have not discussed the ordering within the logical record. Also in this case the criteria seem to be to select characteristics which are permanent and unambiguous for ordering, e.g. the time to which the particular field within the record refers, may be a useful ordering name for fields within a record.

We have argued as if the minimum operation time was the only possible optimization criterion, not minimum operation costs or minimum capacity requirements have implications for the organization within sets? What about the use of direct access contra sequential access storage devices? It does not seem in fact that the capacity criterion is relevant at the present stage of discussion, and we are therefore left with the cost criterion. The relative operating cost may be of importance for the selection of storage device. The minimum operation time criterion will always lead to a preference for the faster device, i.e. usually the direct access device, which, however, is also the more expensive. A compromise may be to choose that medium with the smallest time x cost performance.

5.3 Organization between sets

5.3.1 Organizing the file into sets

The data file will usually be organized into several data sets with independent names. A set will usually

consist of records for units which in some sense are considered as homogeneous. Further, the records will contain data which are related and frequently requested simultaneously. There may also be technical considerations which have determined the organization. The physical capacity of a storage medium may, for example, be one of the reasons for a particular organization.

Sometimes the same data may be duplicated because they are frequently demanded together with two or more data sets, which are rarely requested simultaneously. Even though multiple representations in the data file requires extra storage capacity, it may be an efficient and practical solution considered simply as an investment to save future data processing costs.

5.3.2 Optimum organization of a file into sets

There will always be a number of special conditions related to a particular file which will affect its organization into sets. Some general statements may perhaps still be made about its optimum organization.

Let us, as a starting point only, consider the operation time, assuming that the file is organized in such a way that the search time for the output sets proportional to their size is minimized. If we manage to partition the records of the file into two sets of equal size in such a way that each required output set belongs to either one or the other of the two sets, we have reduced the search time by approximately fifty per cent compared with an un-partitioned file.

Let us assume that the file is divided into two sets, $\mathbf{S_1}$ and $\mathbf{S_2}$, and that we expect that there will be H requests during a given period. We may have the situation that a given output set may belong completely to file set $\mathbf{S_1}$, to file set $\mathbf{S_2}$ or to both of them ($\mathbf{S_1 + S_2}$) and that the probability for each case is $\mathbf{p_1}$, $\mathbf{p_2}$ and ($\mathbf{1- p_1 – p_2}$), respectively. To decide which set a particular output set belongs to, it is necessary to inspect a reference set with one set description record for each file set, i.e. in the example **2** records.

In the above case we may state that the expected operation time in a given period will be:

$$\mathbf{T = 2\,R + R\,.\,p_1\,.\,S_1 + R\,.\,p_2\,.\,S_2 + R\,.\,(1 - p_1 – p_2)\,.\,(S_1 + S_2)}$$

$$\mathbf{= 2\,R + R\,.\,(S_1 + S_2) – R\,.\,(p_1\,.\,S_2 + p_2\,.S_1)}$$

This indicates that we should try to find a partitioning which reduces the probability ($\mathbf{1-p_1 –p_2}$) that the complete file has to be searched.

We may consider the situation in which we form, by duplication of information, a third set $\mathbf{S_3}$

which partly overlaps both $\mathbf{S_1}$ and $\mathbf{S_2}$. If the probability of an output set belonging to this set is $\mathbf{T_3}$, we shall have:

$$T = \mathbf{3\,.\,R + R\,.\,(S_1 + S_2) + R\,.\,p_3\,.\,S_3 – R\,.\,(p_1\,.\,S_2 + p_2\,.\,S_1) - R\,.\,p_3\,.\,(S_1 - S_2)}$$

The expression shows that we shall gain in operation time and the gain will increase proportionally with the probability p and the size difference between the original file ($\mathbf{S_1 – S_2}$) and the set introduced by duplication.

At this stage it seems appropriate to introduce the cost criterion to obtain more realistic conclusions, since the storage cost of the unduplicated file of ($\mathbf{S_1 - S_2}$) records must be less than that of the partly duplicated file of ($\mathbf{S_1 - S_2 + S_3}$) records.

We assume that the search cost is proportional to the search time, and denote the cost per search time unit by **c**. Further, the storing cost is assumed to be proportional to the number of records stored, and we denote by d the cost of storing one record for the time period considered. The expected cost of the situation outlined above will then be:

$$K = \mathbf{c.\{3 + p_1\,.S_1 + p_2\,.S_2 + p_3.S_3 + (1 – p_1 – p_2 – p_3)\,.\,(S_1 + S_2)\}.R + d.\{3 + S_1 + S_2 + S_3\}}$$

The first part represents the variable cost component depending on the expected number of requested **output sets, while the second part is the fixed invariable storing costs.**

Let us introduce the third cost component, the cost of establishing a new set. It may be reasonable to assume that this cost will be proportional to the number of records of the set from which the new set is extracted, and we denote cost per record by **e**.

The change in expected total cost of a given period obtained by dividing the file **S** into two sets, i.e. establishing a new set $\mathbf{S_1}$, having a residual set $\mathbf{S_2 = S-S_1}$, will be:

$$\Delta K = c\,(1 - (p_1 \,.\, S_2 + p_2 \,.\, S_1)) \,.\, R + e \,.\, (S_1 + S_2)$$

The storage cost will not be affected since only a partitioning has been performed. Disregarding the term **c . R**, which will be relatively small, we may say that if the expected cost saving due to the expected reduction in records needing to be searched is larger than the cost of establishing the subsets S_1 and S_2, the partitioning is efficient.

The change in cost due to the introduction of the duplicated set S_3, will be approximately:

$$\Delta K \approx c \,.\, p_3\,(S_3 - S) \,.\, R + d \,.\, S_3 + e \,.\, S$$

which again gives the obvious answer that the cost will be decreased when the sum of the additional storing cost and the cost of establishing the set is less than the cost of the expected reduction in search time.

Let us consider the allocation of sets between two types of storage devices, a fast direct access device with a small c_1 and a large d_1., and sequential access device with a larger c_2 and a smaller d_2. The capacity of the former is assumed to be restricted to keep only one set, while the capacity of the latter is supposed to be unrestricted.

The change in expected cost due to the changing of set S_i to the direct access device, and S_j to the sequential device will be:

$$\Delta K = (c_1 - c_2)(p_i \,.S_i - p_j \,.\, S_j) \,.\, R + (d_1 - d_2)(S_i - S_j)$$

If the two sets are about equal in size, that set which has the larger probability of being requested will be the one which should occupy the direct access device. Considering the reference set as having the probability **1** of being searched, this should always be given priority to the direct access device.

The above model may be generalized to the case in which we have **M** file sets from which may be formed **N** additional logical sets by combining pairs of file sets, triples of file sets, etc. The general cost function will be:

$$K = c\,(M + \Sigma^{M+N} p_i \,.\, S_i) \,.\, R + d\,(M + \Sigma^{M} S_i)$$

with the addition of a cost of establishing new sets, if this is done.

The additional total cost implied by the division of any existing file set S_i into two file sets S_i**'** and S_i**"=(**S_i**-**S_i**')** will be:

$$\Delta K = c \,.\, (1 + p_i' \,.\, S_i' \,.\, p_i'' + S_i'' + (p - p_i' - p_i'') \,.\, S_i - p \,.\, S_i) \,.\, R + e \,.\, S_i$$

$$\approx - c \,.\, (p_i' \,.\, S_i'' + p_i'' \,.\, S_i') \,.\, R + e \,.\, S_i$$

The partitioning will be efficient when the saving in search cost implied by the partitioning is larger than the cost of dividing the set S_i into S_i**'** and S_i**"**.

The introduction of a duplicated file S' will give the change in the expected cost corresponding to:

$$K = c \,.\, (1 + p_i' \,.\, (S_i' - S_i)) \,.\, R + d \,.\, S_i' + e \,.\, S_i$$

where S_i represents the set from which the requested output was previously retrieved assuming that $S_i < S_i'$.

If the expected reduction in search cost is larger than the sum of the cost of establishing a subset of S_i and the additional cost of storing this duplicated set, the duplication is efficient.

If we have limited capacity of a fast but expensive direct access storage device with operation and storage costs per record c_1 and d_1, respectively, and a less expensive sequential storage device with costs per record $c_2 > c$ and $d_2 < d$, we may consider moving set S_1 from the sequential to the direct access store in exchange with the set S_2.

The expected change in the file cost implied by this set exchange will be:

$$\Delta K = [\,(c_1 - c_2) \,.\, p_1 \,.\, R + (d_1 - d_2)\,] \,.\, S_1 - [\,(c_1 - c_2) \,.\, p_2 \,.\, R + (d_1 - d_2)\,] \,.\, S_2$$

and the exchange will be an efficient re-allocation when the change is negative, e.g. when $S_1 = S_2$ and $p_1 > p_2$.

A similar consideration applies in connection with the problem of which sets should be kept on an on-line medium and which should be stored off-line.

To sum up this discussion, we may conclude that when no more efficient partitioning, duplication or re-allocation can be made, the optimum organization of the file into sets has been obtained.

6. Summary of the data file system

The data file system as discussed in the present paper consists of two parts, the file sets and the file procedures.

The file sets are of two kinds, the reference sets and the data sets which are all related through a hierarchy structure. The reference sets comprising all non-terminal sets are vocabulary or catalogue sets while the datasets which are the terminal sets are either record or table data sets. Each set has a symbolic name which determines the branch through the set hierarchy from the set root or master set. The real set name is only known to and used by the internal system.

The sets may either by open for common use or locked for unauthorized use. The branch to a locked set goes through one or more security sets which require secret passwords for further branching. Without the knowledge of these passwords the wanted sets will be inaccessible.

The file procedures are the working instructions for the file system which may work either in the storing, the retrieval or the deleting mode. The backbone of the file procedures is the master procedure which takes care of the main parts of the file administration and ties the other procedures together in a meaningful pattern.

The organization of the file raises two more problems. The first is the organization of records within the sets. The most appropriate arrangement will probably be an ordering of the records according to unit identification, since an important purpose of the establishment of the file will be to be able to link data from different fields and periods together at a unit basis. The second problem is the organization of the file into sets by partitioning and duplication and by allocation of sets among different types of storage. The answer to this problem may be an analysis of expected utilization and cost conditions.

7. Data representation

The data may be stored in different ways in the file. We shall here outline two different forms. The first may be denoted as the situation set approach, and the other, the change set approach.

In the file constructed by situation sets, each set describes the situation at a certain point of time, to which all observations refer. This does not mean that only status or stock characteristics can be included. The situation at the end of the year may very well be described by the value or quality of the the production during the last twelve months. The situation set approach has the advantage that by extracting data from a series of situation sets we get complete and syncronous time series for the different characteristics, even for those which have not changed. But since we cannot establish situation sets too frequently, we shall not be able to be precise in time specification.

In the change approach on the other hand, the characteristics are recorded only when they change. The birth date will therefore only be recorded once for each person, while the value of production per calendar year of an establishment will normally change each year, and will therefore also be recorded each year. The advantage of the change set approach is that we only need to store new data and that we may specify the point of time for the change precisely. The drawback is that we shall not have commonpoints of time at which we can compare the different characteristics.

The most convenient data representation will probably be a file which may comprise both types of sets, this will be a file in which data duplication will be an important aspect.

8. Automatic file construction in Norway

In the Central Bureau of Statistics of Norway, Implementation of file systems is now going on. To illustrate some of the ideas discussed above, we shall give an outline of the more important work carried out [**24**].

8.1 Population files

8.1.1. General information

The population registration in Norway is authorized by a law of November 15, 194.6, and the Central

Bureau of Statistics has been charged with the responsibility for acting as a central office for the registration. Today there are 454 local registration offices in Norway. In addition to the register documents required by the law, there are punched card registers at each office. The registers serve a number of non-statistical administrative purposes as well as being important instruments for the current population statistics.

In connection with the I960 Census of Population, a central population register to serve the need for a nation-wide system of permanent identification numbers was discussed. The Ventral Bureau of Statistics worked out plans for such a register and was charged with the implementation of a nation-wide register in 1963.

The identification number constructed for each person is an 11-digit decimal number. The first 6 digits, called birth data, represent date of birth with the omission of the two century digits of the year, the next three digits are used to distinguish between persons with the same date of birth, between men and women, between people born at the same date in different centuries, etc. The last two digits are check-digits computed according to the modulus 11 method.

The establishment of the central register was done by punching name, address, sex and date of birth for all persons from the census lists. By an automatic routine identification, numbers were computed and assigned to these cards. This routine keeps an account of which numbers are free and which are occupied for any given date of birth. The numbered cards were distributed to the local offices for control and for acquainting the local offices with the numbers to be used.

Births and -Immigrants are reported currently to the central office which assigns numbers to them. Because of the time difference between November 1, I960, when the census was taken and October 1, 1964, which was the date chosen as a check-point for the loca offices, the control and supplementing work became extensive. The central register is held on magnetic tapes, and the Norwegian population of less than £ millions requires at present about 30 reels of tape. During the period of establishment and control, several hundred reels have, however, been used at the same time.

The central register will be kept up-to-date through the regular channels for registering births, marriages, migration and deaths. From January 1, 1967, the identification numbers are used all over the country in tax and social insurance work. The police as well as the health authorities will probably introduce the system in their work, and we hope it can be introduced in the school system as early as the elementary school level.

Through the introduction of central population register systems in statistical and non-statistical data collection, a large collection of data related to the individuals and identified by the general identification numbers are obtained. This mass of data can be ordered and linked by persons to a much larger extent than was previously possible. We can follow the demographic data for all individuals from the I960 Census of Population. From 1964-, the relationships between parents and children are recorded on machine accessible media. Such relations permit the association of a large number of indirect characteristics to each unit. A child may, for example, be characterized by actions or states related to its parents. The registrations of births and deaths will also be related to a number of medical data. Data already being stored give new possibilities for the analysis of the conditions determining variation and extent of births, marriages and deaths. In the same way, migrations may now be studied much more realistically than the previous data permitted. All these facts will have great implications for the planning of the 1970 Census of Population.

The data files will also store the income and tax data for individuals obtained from the Tax Authorities. By means of the common identification numbers these economic characteristics can be linked with data from the demographic files, and the economic analyses may be carried out against a much more detailed background. But we shall also be able to take up studies of births, marriages, etc. explained by economic factors.

Later on, data from a number of other fields, all collected by using the central population register or copies of it, will be added to the files. These data, including social conditions, education, etc. will open up a new field for socio-economic studies in which the interaction of sociological and economic conditions can be studied at a unit basis.

8.1.2. Technical description

The population file will consist of a number of different sets, of which we shall here discuss those which now include the demographic data.

There are two kinds of sets, those describing a situation at a certain point of time and those describing changes registered during a certain period. A situation set is physically a combined register and data set, with one record for each person registered. The record ordering is by identification number. Each record has a fixed length of 168 characters and the following content:

Identification number (including birth date)

Name

Street address / other equivalent address

City, etc.

Municipality id.no.

Status of existence (living in the country, emigrated, dead)

Status of marriage

Id. no. of father

Id. no. of mother

Date of id. no. assignment

I960 census data

Date of immigration

Date of death

Certain technical data

The first mentioned field contains the permanent identification which is also the version used for internal reference, while the next three fields represent the most frequently used external identification. Since the latter, however, may change over time, it may also be considered as data.

A set of changes consists of variable length records, each named by identification number and arranged in the same ordering as the sets of situation. A logical record of a set of changes contains the following fields:

Identification number

Date for registration of change

Type of change (real change / correction)

1. position of the field to be changed

Number of positions to be changed

Change

Several logical records may be combined in one physical record.

It is important to note that there is a very definite relation between situation sets for two points of time and the change sets for the intermediate period.

8.1.2.1 Identification number system[1]

Since the introduction of a common number was considered an important task, much effort was put in the selection of the numbering system.

The number should be processed by technical equipment and had to satisfy the restrictions of both punched card equipment and desk calculations. This reduced the possible character range to decimal digits.

The easiest way of giving identification numbers to n individuals in a group is to give the first individual the number 1 and the last individual the number n using all numbers from 1 to n, paying no attention to which individual is becoming the first and the last. The question was whether the population should be considered as one or several groups in the numbering process and how the persons should be grouped. If more than one group should be used, the groups should be given numbers and the group number should be a part of the identification number. Since the identification number should be permanent and follows the person through his lifetime, the grouping must be based on non-changeable data. Among the data recorded in the register only the date of birth and sex could be used for such grouping. Sex and date of birth were considered vital information which could be collected with high accuracy. In Norway only a maximum of **250** persons have been born on a single date. This fact makes it possible to identify each person by a **3** digit number, and this number has capacity to distinguish between males and females and between persons born in the 19th and the 20th century. The identification number therefore consists of **9** digits:

Date of birth: day, 2 digits month, 2 " year, 2 "

Serial number: 0 - 74-9, 3 digits, 0 - 4-99 for persons born in the 20th century, century and 500 - 74-9 for persons born in the 19th century.

Odd numbers for male, even numbers for female.

This solution with the addition of **2** check digits, was chosen for the identification number. The main reason for the choice was the very short number of only 5 digits which had to be added to the date of birth. One did not wish to burden the public with a longer number than necessary. This solution has two disadvantages:

a. It gives a complicated number generating routine which can hardly be decentralized.

b. The identification number contains information about the individual, date of birth and sex, and has to be changed when an error in sex and date of birth is detected.

All identification numbers are generated by a machine routine, and a tape catalogue keeps track of the next number available within each date, sex and century. With the batch processing techniques used it is impossible to run the number generating routine more often than a few times a week. A set of punched cards with **2** available numbers for each date is therefore maintained. This makes it possible to give identification numbers to local registration offices by telephone in special cases. When a number is used from such a card, the card is sent to the machine and a new card with the next available number is punched the next time the number generating routine runs.

Until now **130 000** errors have been corrected in date of birth and sex. In each case a new identification number has been generated. This is a somewhat higher percentage than originally expected.

The alternative to the identification number chosen was a numbering within one group. A serial number running from **1** to **9 999 999** would have been sufficient for the present and future population in at least **50** years. This gives only **7** digits plus check digits, but to the users it might have been felt as a longer number than the present number. The advantages by such a number would have been:

a. A simple number generating routine with possibilities for decentralized number generating.

b. A number free of information about the individual and no need for generating a new number

[1] **Sections 8.1.3 – 8.1.7 are extracted from a psperr on Technical Problems of Setting up and Maintaining a Population Register by Computer prepared by Mr. Erik Aurbakken, Central Bureau of Statistics of Norway.**

when date of birth and sex are corrected.

It was taken for granted that one or more check digits should be added to the identification number to protect it against misinterpretation and transcribing errors. It was known that when date of birth is handwritten approximately **1** per cent of the dates shows to be in error. These errors may have a special structure and it could not be taken for granted that a modulo **11** check digit with standard weights such as presented by IBM would be the best solution. A representative sample of errors in date of birth was collected and a study was carried out to select the most effective set of weights in the modulo **11** formula. It proved possible to find a set of weights which was more effective than the IBM standard set. It was then decided to use **2** check digits, the first digit calculated by a selected set of weights and the second digit calculated by the IBM standard weights. The second check digit includes the first and can be calculated and checked by an IBM punching machine. The first check digit has to be checked by the computer. With these two check digits we have estimated that approximately 3 errors out of **10 000** will be left undetected.

8.1.2.2 Establishment of the register

A card was punched for each person with municipality of residence, date of birth, sex and name. The cards were converted to magnetic tapes, sorted on date of birth and the identification numbers generated and attached. The tapes were sorted alphabetically by name within each municipality and a new punched card was produced in this order and sent te the local registration offices. The local offices were asked to check the cards against the situation in their registers at the **1 October 1964**, copy the identification numbers to their own register and return the cards to the Central Bureau with necessary corrections and the present address and marital status of the persons. If a person had migrated to another municipality his card was sent to this municipality before it was returned to the Bureau.

From **1 October 1964** the identification number was taken into use by the local registration offices. All reports on migration, marriages, etc. to the Central Bureau of Statistics and between the local offices should contain the identification number to make it possible to update the central register.

When the cards were returned from the local offices to the Bureau all errors reported were corrected and the address and marital status were punched. If sex or date of birth were corrected a new card with a new identification number was punched and returned. All cards were matched against the file used in the generation of identification numbers to check for completeness. Missing cards were copied from the tapes and sent to the local office in question.

The use of printed cards as a medium for distributing the identification numbers to local offices proved to be a success. The cards could easily be sent from one local office to another and were used in mechanical routines by local offices with punched cards registers. Relatively few cards were lost or damaged during the process.

8.1.2.3 Register routines

The main principles for the checking routines are the following:

a. Errors shall be discovered at the earliest possible stage

b. The routine can be run at any interval, a new run with new data can be started before errors from the
previous run are corrected.

c. All corrections of errors are checked.

d. The checking of data cannot be performed in one routine or programme. Different methods must be used to report the errors as soon as possible. One may distinguish between the following checks:

e. Manual inspection of reports.

f. Control of check digits on the punching machines.

g. Automatic checking of each report (checking codes for validity, logical combinations, duplicates).

h. Checking the chronology of reports, new values against old values.

i. Checking of programming errors and machine malfunctions.

Some remarks will be given on category c, d and e.

Each type of reports needs a special programme for checking. Some of the errors can be corrected in the Bureau, these are sorted in the same sequence as the reports are stored and printed. Other errors must be handled by the local registration offices, these errors are sorted alphabetically on the name of the persons. All errors are given a reference number with one check digit and this number is tised when corrections are punched. Some data are collected for statistical purposes only and are not recorded in the register, but all data are checked in the same rountine. This had caused some delay in the processing of the statistics the first year.

All reports are dated by the local registration offices and will normally be received by the central office in chronological order. However, reports may be delayed and this order may be broken, reports containing errors will normally be delayed up to several weeks. It is therefore necessary to check that the reports are handled in chronological order for each person in the updating routine. It has proved necessary to keep a date for each field in each record in the situation file to know when this field was changed last time. If a report reaches the updating routine out of sequence an error is signaled and the report is manually inspected, the explanation may be an error in the date or the identification number or that the report really was delayed. Delayed reports will be recorded in the change file and have no effect on the situation file.

When the data reach the updating routine, all errors should have been corrected and the corrections checked. As an extra safety some checks are repeated in the updating routine.

The first two years the error rate has been higher than estimated, especially in the identification number. More information and training of the local staff will probably give better results.

Different problems have been encountered in constructing an updating routine for the register. The problems are of two different categories:

a. Problems in connection with the fact that the reporting system from the local to the central registration office cannot be adjusted easily and modified to meet the demand of a central register stored on magnetic tapes.

b. Problems in connection with the fact that the register shall serve two purposes: Show the most up to date situation of the population and be a source of data for statistics.

The first category of problems we will have to live together with for some years still. One problem is that the same information may be reported from more than one source, children born are reported both from the hospital and from the parson and if the parents are dissenters we may get a third report. Reports about the same child may come from different municipalities, children are often born outside the municipality where the parents are resident.

The reporting system has to be modified in co-operation with different ministries and different laws must be altered. Work is going on in this field at present.

The population file shall store all data collected about a person. The information has been organized in two sets, the situation set and the change set. The situation set has one logical record for each person and shows the most up to date information about each person and can be used for printing lists and addressing purposes. Each field in the situation set has a date showing the date this field was changed. The change set has one logical record for each change and contains all changes earlier performed in the situation set. Each change is dated and the set can be used to reverse the situation set, make it show the situation at an earlier date. The changes are sorted in chronological order, in ascending order to bring the situation forward in time and in descending order bringing it backward.

The updating routine handles three types of transactions: new units, changes to existing units and corrections.

New units are children born and immigrants. These transactions pass a number of generating routine before entering the updating routine. Immigrants are matched against earlier emigrants to protect against duplicates in the register, the emigrants are stored in the register as emigrants after leaving the country.

Changes are matched against the situation set, the old value is removed and stored as a record in the change file and the new value substitutes the old one.

Corrections are used to correct errors in the register discovered at the local or central registration offices or by users. Corrections are processed in the same way as changes in the updating routine, but when the situation set is reversed by the old changes corrections have no effect on the new situation. We have found it necessary to distinguish between two types of corrections: ordinary corrections and conditional corrections. Conditional corrections are only performed if a given condition is satisfied in the situation set. The need for conditional corrections is due to the delayed batch processing method used. This method does not give immediate access to a record for the staff. The problem can best be illustrated by an example. A user reports an error in the name of a person. This can be corrected by a correction transaction, but meanwhile this person may have changed name by marriage. If the report is processed before the correction, the name will be changed back to the old name. In this case it will be safe to make the correction conditional to assure that the name will only be changed if it has the wrong value.

In principle all statistics should be compiled from the situation and change sets. When reaching the updating routine the data have passed all checks and all corrections have been performed. Until now it has not been possible to keep these sets sufficiently up to date and the date for the statistics have been extracted at an earlier stage. The present goal is to speed up the routines for the register so that the data can be fully corrected before extracted to the statistics. When a complete integration is reached, (the statistics compiled from the situation and change file) some information, such as sex and marital status, can be extracted from the situation set instead of being collected and punched from the reports.

8.1.2.4. Organization of the file

The data are organized in two main sets, the situation set and the change set, as earlier described. The change set can later on be divided in subsets, each subset covering a period of time.

The situation set is kept in ascending order by identification number in the updating routine and only this set is updated. Different lists have to be produced in other orders, by name within municipality, by date within municipality, by name within date and perhaps other orders. This involves, in our present routines, a heavy load of sorting. To reduce the computer hours used for sorting it may be necessary to construct routines for updating the register in different orders. The optimal solution depends, among other things, upon the frequency of listing the different orders. This frequency is not known at present.

Until now all data have been duplicated in the different sets. This may prove to be unnecessary. If some data are not required in some of the lists, this data should be left out in the set kept in this order.

Conventional batch processing methods are used in our routines. On line processing methods have not been investigated. The available computers do not offer sufficient capacity for an on line solution of a problem of this size.

8.1.2.5 Service to other agencies.

From the first of January 1967 the identification number was taken into use by three other institutions, in the taxation routine for personal taxpayers, in the social insurance administration and by the Directorate for Seamen where it will be used in different administrative routines.

The Central Bureau has produced lists and punched cards to inform these external users about the identification numbers. All corrections in identification numbers must be distributed regularly to the external users, and they report back about different discrepancies.

The central population register only covers persons situated or earlier situated in Norway. The other registers also cover other categories, taxpayers and seamen may be foreigners and some of them may be covered by the social security scheme. Preliminary this problem has been solved by introducing special series of identification numbers to categories not covered by the central population register. Foreign taxpayers have got a series of numbers with 99 as the 2 first digits and date of birth has been excluded from the number for these persons. The Directorate for Seamen has kept the date of birth for all persons and has chosen to add 4-0 to the day in the date for all foreign seamen and by

this established an identification number analogous to that of the central population register. If these solutions do not work satisfactorily, the final solution may be to include these categories of foreigners in the central population register in the ordinary series of identification numbers. Before doing this one must assure that the **3** digits used to id,entifiy persons within one date of birth have sufficient capacity to expand the register by these categories.

8.1.2.6 Final remarks

During the work with the register we have realized that producing statistics and maintaining a register are two different things. Experience and methods in one field cannot easily be adopted to the other. This led us to greatly underestimate the problems and the resources needed to solve them. This had to be paid for by preliminary solutions, delays in processing and extra costs in the future.

Of course, our solutions have been strongly influenced by the situation at the time the work was started, especially concerning the existing routines and the available technical equipment. There was a strong demand for a common identification number system and we had sufficient data collected by the I960 Population Census to start establishing a register. This was enough to get the work started. However, the time laying between the census and the establishing of the register was 4. years and this caused extra problems and workloads in the first period, and we still have to work hard to get the register sufficiently up to date.

The next years will be used to improve the reporting system from the local to the central registration office to facilitate an effective flow of information from the local to the central registers with a minimum of duplication of written reports. This will of course influence the data processing routines which will have to be modified or reconstructed.

8.2. Files for establishment and enterprise data

8.2.1. General information

In connection with the 1953 Census of establishments, a system of central registers for establishments and enterprises was established. These registers have been maintained and kept on punched cards until 1965. They comprise economic activities in most industries except for Agriculture, Forestry, Hunting and Fishery. In some industries the registers do not include enterprises in which the owners are working alone. The register of enterprises contains about **110 000** units, while the register of establishments covers about **130 000** establishments within the same enterprises.

Up to 1965, the identification number system for enterprises was constructed in such a way that when the enterprises were sorted in ascending order, they were automatically also in alphabetic order. This required open space between any two numbers in case a new enterprise with a name alphabetically between the two should appear. The establishments were numbered by adding a two digit number after the enterprise number. This system had serious disadvantages and its advantages were subject to the use of punched card equipment.

To satisfy the requirement for permanent identifiers, independent numbers for enterprises and establishments were introduced in 1965. The identification numbers are assigned continuously with no built-in information. They consist of **6** digits plus 1 check-digit computed by the modulus **11** method. The check system will fail to detect **1-2** out of **100** errors in the numbers. This is assumed to be insignificant because these numbers in general will be pre-coded on questionnaires by machines in most applications and the probability for errors is therefore expected to be very small. The relation between the establishments and the enterprise to which they belong is taken care of by using the enterprise number as a characteristic of the establishments.

The introduction of permanent identifiers was made possible by transferring the register processing to tapes which allow alphabetic sorting directly by name.

This reorganization of the register has initiated an important discussion about definitions of "births", "migration" and "deaths" for both enterprises and establishments. The two units may have different life-time and -pattern, even though the latter may belong to the former for a certain period. This is one of the main arguments for using independent numbers for an enterprise and its establishments.

A difficult problem in connection with the maintenance of these registers is to get information about new units fast enough. Many sources of information, of which the social insurance administration is important, are used. Maintenance is also carried out through information collected by Name Cards. These cards are very simple questionnaires, partly filled out with information from the registers and distributed to enterprises and establishments which are asked to correct and supplement the information if necessary, and return the cards to the Bureau.

The main use of these registers has been in connection with statistical surveys, but the registers have also proved to be of great value for industry itself. There are, however, very important non-statistical administrative needs which cannot be served by these registers in their present form. These needs and their possible solutions will be discussed in a section below.

In connection with the next Census of Agriculture, the incorporation of units within this industry will be considered.

The introduction of permanent identification numbers in the registers of establishments and enterprises will give us data files for these units from the 1963 Census of Establishments. For a smaller number of selected enterprises, the files are now being extended backwards to 1959.

These data files have already proved to be an excellent source for fast computation of statistics describing the sizes of different industrial groups, etc. When more annual data with permanent identifications of units are stored, we shall be able to intensify the automatic control procedures used for editing collected data, to compute the age distribution of establishments and enterprises, and the associated probabilities for "deaths". Further, we shall be able to compute the probability for the "birth" of a new unit based on the conditions in the different fields of the economy.

So far, many micro-computations have been carried out to study the structure of production functions. The data files will also allow us to take up micro-computations which also include dynamically formulated hypotheses, in which the production, investment, price policy, etc., are assumed to depend on the individual establishment's or enterprise's historic success, as well as on the more common market factors.

In general, the data files for establishments and enterprises will include data on production, consumption of raw material, employment, accounting etc. As will be discussed below, we hope, however, that linkage between data of the population files and data of the files of enterprises will become possible in the future.

8.2.2. Technical description

The files of data for establishments and enterprises will also contain many types of data sets. In the same way as for the population files, we shall restrict the discussion to the sets most directly related to the registers.

The registers of establishments and enterprises are merged in one set which is a combined register and unit data set. Each unit is represented by a fixed length record of **152** characters. The records are arranged according to the unit identification numbers and contains

Identification number

Name

Street address

City, etc.

Unit type (establishment, enterprise)

Id. no. of owning enterprise

Municipality id. no.

Industry code

No. of employees

Size of gross income

Ownership code

Certain data for technical use

This set is also a situation set and refers to the situation at a certain point of time. The register part of the set is represented by the first four fields of which the first may be considered as the permanent and internal identification, while the following three fields refer to the external identification which may change.

There are also change sets which refer to changes registered in specified periods. A change set consists of fixed length 64 characters records arranged by identification number ordering. Each record consists of:

Identification number

Date for updating situation set

Change code

No. of the field in situation record to be changed

Change

A similar relationship as in the case of population files exists also by the situation sets and change sets described here.

8.3. General purpose register system

Tax authorities and social insurance administration need a nation-wide register of employers. The units of such a register will be owners of enterprises, which are partly individual persons and partly companies. The register of enterprises has been considered, but because its extension is limited, it has been rejected. A proposal that the Central Bureau of Statistics should be asked to establish and maintain a register of employers based on reports from local tax and social insurance offices has been made.

From a statistical point of view this proposal is of great interest because it would supply important information for the register of enterprises.

The Central Bureau of Statistics has therefore outlined a solution which implies a consistent general purpose system of registers which can be used for registration of population, companies, enterprises, employers, employees, tax payers, etc., and connected to derived registers of households, establishments, etc. Within this system any person or company will be identified with the same number whether he or it appears as a person or company, employer, employee or tax payer. Also the enterprises get identification numbers corresponding to their owners'.

The system may be obtained by first establishing a register of companies or juridical, non-physical persons as an extention to the present central population register, and with an identification number structure corresponding to that used for persons. Thus, the present numbers of enterprises are exchanged with the numbers of their respective owners.

8.4. Files for regional data

8.4.1. General information

Norway is divided into 460 municipalities, to which important regional data are associated. To obtain a higher serviceability, a register of these municipalities is kept on a machine medium. The size and borders of each municipality are changed from time to time, and the number of units in, and the composition of the register also changes.

Data published for each municipality are stored as tables in the file. These data comprise the name of the publication, the name of the table, the text for each column and the table matrix, each row of which preceded by the identification number of the municipality. The data are stored in such a way that each row of the published table may be regarded as an independent table, i.e. each published table is divided into as many single row tables as there are municipalities.

The file of regional data may be used for a special information service, for special periodic regional reports, for historic surveys of the individual municipality, as well as giving a basis for general research in regional development and structure.

8.4.2. Technical description

The register and the file for municipalities are punched on **7** card types:

Card type 1: Register cards
" " 2: Publication name cards
" " 3: Table name cards
" " 4: Column text cards
" " 5: Format cards
“ " 6: Row cards
" " 7: Footnote cards

Each register card contains an entry with the official name of the municipality and an exit with a four digit identification number for the municipality. The identification number is equal to the code used in all official statistical publications, by the Post Department and other government agencies. The card also includes a time specification which determines for which period this particular card is a valid register card, i.e. this means roughly speaking that a new version of register cards is established for each period. The time specification may also be considered as a part of the external name.

The publication name cards, table name cards and column text cards may be considered as a hierarchy of table description sets. The publication card contains the name of a publication and year of observation as its entry. The exit is a code which is the real name of the set of all table name cards belonging to this publication.

The table name card includes the name of a table as its entry and refers by a real name to the set of all column text cards and row data cards of the table. Each row card may be considered as a table data card with the symbolic name "publication name, table name, name of municipality".

The data set may be further subdivided by the column cards. The column card has the column text as its entry and refers to the data set within that particular column.

Footnotes are taken care of in two ways. Footnotes referring to the whole table are column-included in the table name card or qplumn text card, respectively. Footnotes referring to an individual row or field within a row are punched in footnote cards which may be considered as appendices to the respective row cards.

The term "card" used should be interpreted as a logical rather than a physical card because it is a record which may very well extend to several ordinary punched cards. The table name may, for example, occupy several cards, as may also the publication name, the column text, the table row, and the footnotes.

9. Final remarks

In this paper we have concentrated on the file aspects of a statistical information system. Many problems, ideas and solutions have been presented, not all of which may be expected to be useful, but it is hoped that they may be a basis for further discussions.

It should also be emphasized that there are many and important aspects, which are connected to those discussed, which are left for future discussions. First, the statistical files as discussed here will without doubt have important implications for the collection schemes. A revaluation of the collection schemes on the basis of the file system may result in a rejection of the current practice in favour of other schemes. This may result in the development of modified or new collection methods and techniques.

The implementations of ideas like those presented in this paper for the data processing system will, of course, raise special technical problems which, however, may be dependent on the model and configuration of the particular equipment available to such an extent that they may be most efficiently discussed in smaller groups.

The files organized as outlined in this paper, will open up new fields for statistical service and analysis,

and the re-formulation of problems in the light of the new possibilities will result both in large and interesting tasks, which may engage the statistical offices to an increasing degree in future years.

Reference

[1] C.F. Balz and R.H. Stanwood: *Literature on Information Retrieval and Machine Translation*, International Business Machines, N.Y. 1966

[2] R.C. Bose: *Error Detecting and Error Correcting Indexing Systems for Large Serial Numbers*, Paper presented at the 35th Session of the International Statistical Institute, Beograde,1965

[3] Central Data Corporation*: INFOL, General Information Manual*, CDC, Col., 1965

[4] I.M. Chakravarti: *On the Construction of Difference Sets and their Use in Search for Orthogonal Latin Squares and Error Correcting Codes*, Paper presented at the 35th Session of the International Statistical Institute, Beograd, 1965

[5] W.A. Clark: *Data Management*, IBM Systems Journal, Vol.5, No.l, 1966

[6] R.C. Daley and P.G. Neumann: *A General Purpose File System for Secondary Storage*, Proceedings from Fall Joint Computer Conference, 1965

[7] E.S. Dunn, Jr.:*Review of Proposal for a National Data Center*, Bureau of the Budget, Washington D.C. 1965

[8] E.v Hofsten: *Population Registers and Computers - New Possibilities for the Production of Demographic Data*, Review of the International Statistical Institute, Vol.34., No.2, Haag 1966, pp. 186-194

[9] B. Langefors: *Information Retrieval in File Processing 1*, BIT, Vol. 1, No.l, Copenhagen, 1961, pp. 54-63

[10] B. Langefors: *Information Retrieval in File Processing 2*, BIT, Vol. 1, No.2, Copenhagen, 1961, pp. 103-111

[11] B. Langefors: *Some Approaches to the Theory of Information Systems*, BIT, Vol. 3, No.4, Copenhagen, 1963, pp.229-254-

[12] B. Langefors: *Information Systems Design Computations Using Generalized Matrix Algebra*, BIT, Vol. 5, No.3, Copenhagen, 1965, pp.96-121

[13] S. Lebergott: *The Accounts and the Computer*, Ninth General Conference of the International Association for Research in Income and Wealth, Lorn, 1965

[14] G. Nathan: *On Optimal Matching Processes*, University Microfilms, Inc., 1964

[15] G. Nathan: *Outcome Probabilities for a Matching Process with Complete Invariant Information*, Central Bureau of Statistics, Jerusalem, 1965

[16] S. Nordbotten: *A Statistical File System*, Statistisk Tidsskrift, No.2, Stockholm, 1966, pp. 99-109, Translated version of: *Elektronmaskinene og statistikkens utforming i arene framover*, Det Nordiska Statistikermjzfte i Helsingfors I960, Helsingfors 1961, pp.135-141

[17] S. Nordbotten and T. Aastorp: *On Library Routines in Statistical Data Production*, Statistisk Tidsskrift, Stockholm, 1962, pp.166-174

[18] S. Nordbotten*: On Statistical File System*, to be published in No.2 of Statistisk Tidsskrift, Stockholm.

[19] I. Ohlsson: *National Accounts as an Instrument for Co-ordinating Statistics*, Ninth General Conference of the International Association for Research in Income and Wealth, Lorn, 1965

[20] G.H. Orcutt: *Statement in: A compendium of Views and Suggestions from Individuals*, Organizations and Statistics Users, Subcommettre on Economic Statistics, Congress of DS, Washington D.C., 1965

[21] S. Rokkan, ed.: *Data Archives for Social Science,* Publications of the International Social Science Council, Mouton and Co., Paris, 1966

[22] N.R. and N. Buggies*: A Generalized Economic Retrieval System and Data Files*, Paper presented at the ASA Regional Meeting, Harvard University, Mass., 1966

[23] N.R. and R. Ruggles: *A Generalized Economic Information Retrieval System and Instruction Manual for Information Input*, Economic Growth Center at Yale University, Conn., 1966

[24] Ths. Schl^tz: *Use of Computers in the National Accounts of Norway*, Ninth General Conference of the International Association for Research in Income and Wealth, Lorn, 1965

[25] T. Seppäla: *On Optimization of Maintenance of a Register*, BIT, Vol.6, No.3, Copenhagen, 1966, pp. 212-227

[26] G.S. Tootill,ed.: *IFIP - ICC Vocabulary of Information Processing*, North-Holland Publishing Company, Amsterdam, 1966

[27] B.C. Vlckery: *On Retrieval System Theory*, Butterworths, London, 1961

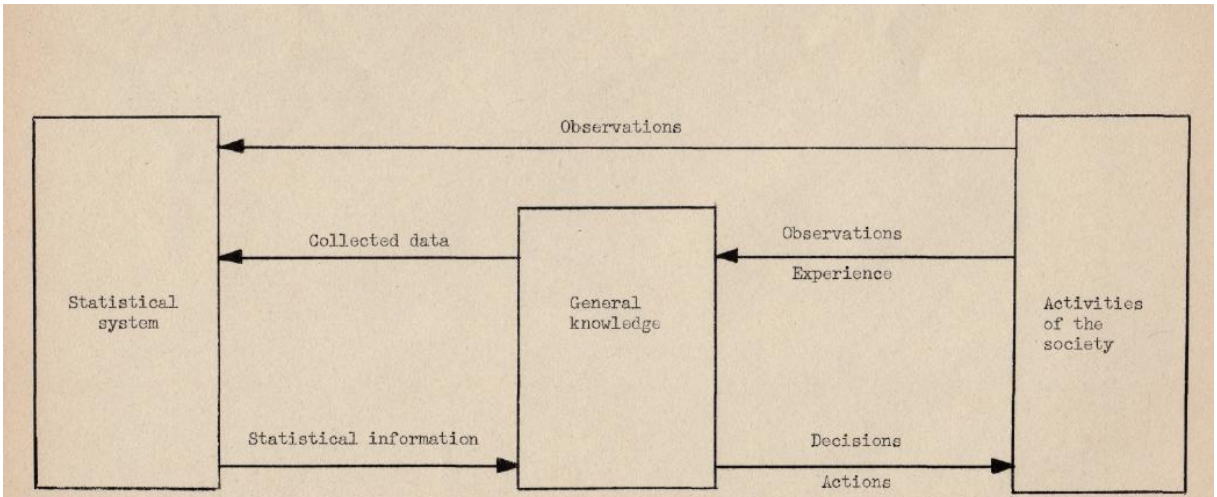

Figure 1: The statistical system and the society.

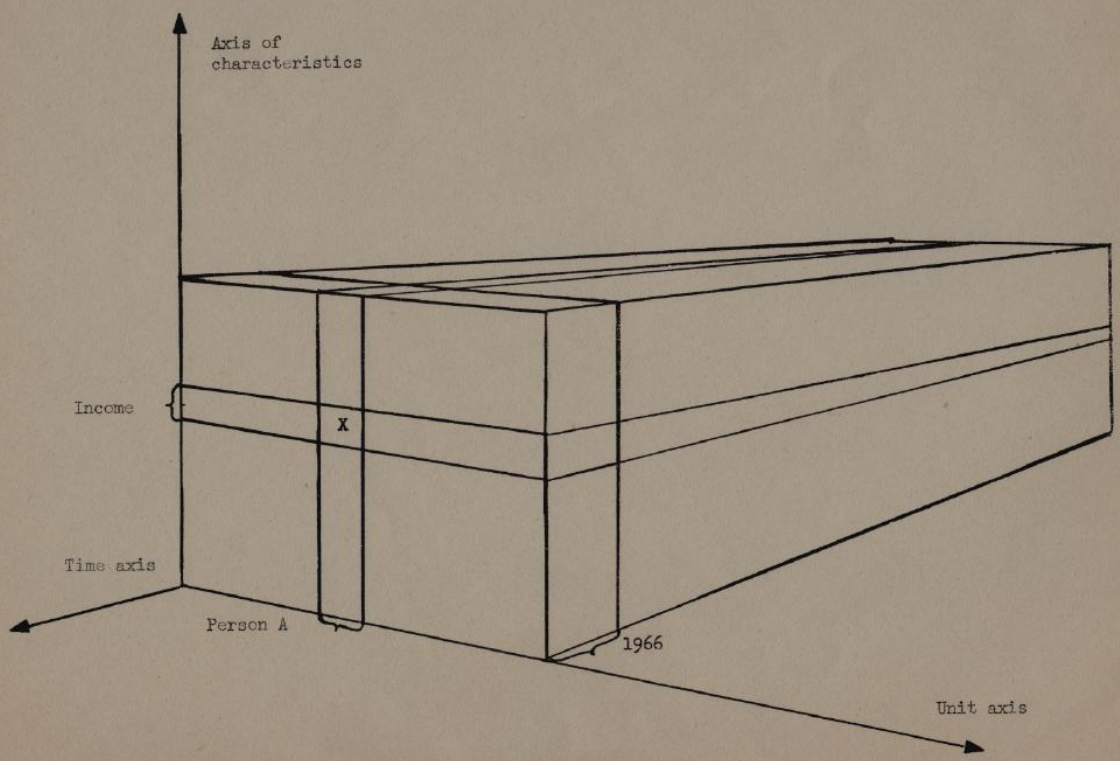

Figure 2: The data box and the storage of income X for person A in 1966

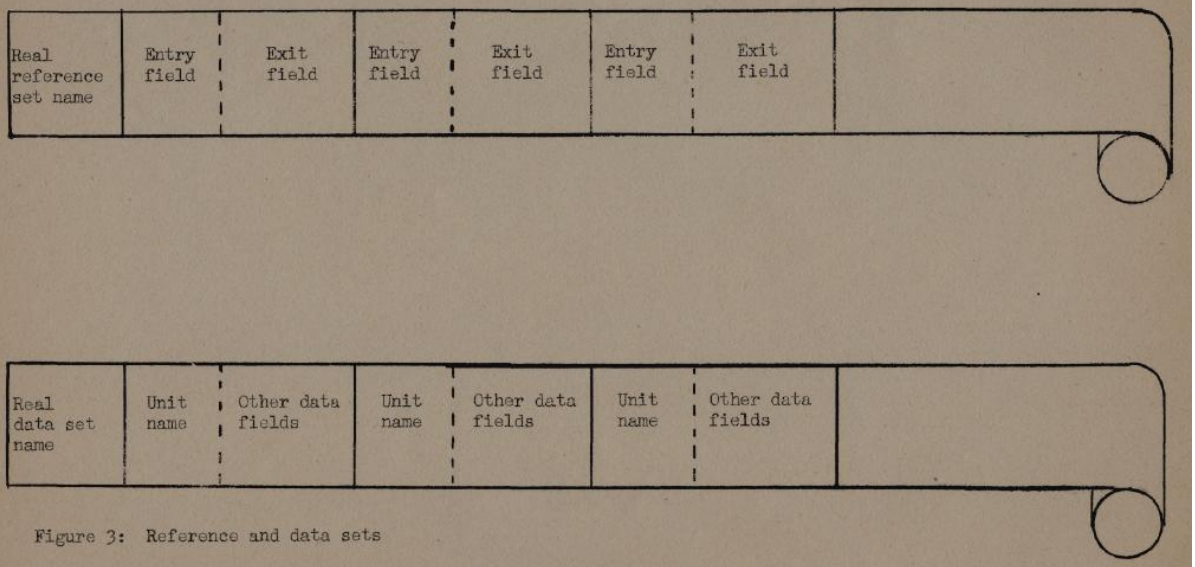

Figure 3: Reference and data sets

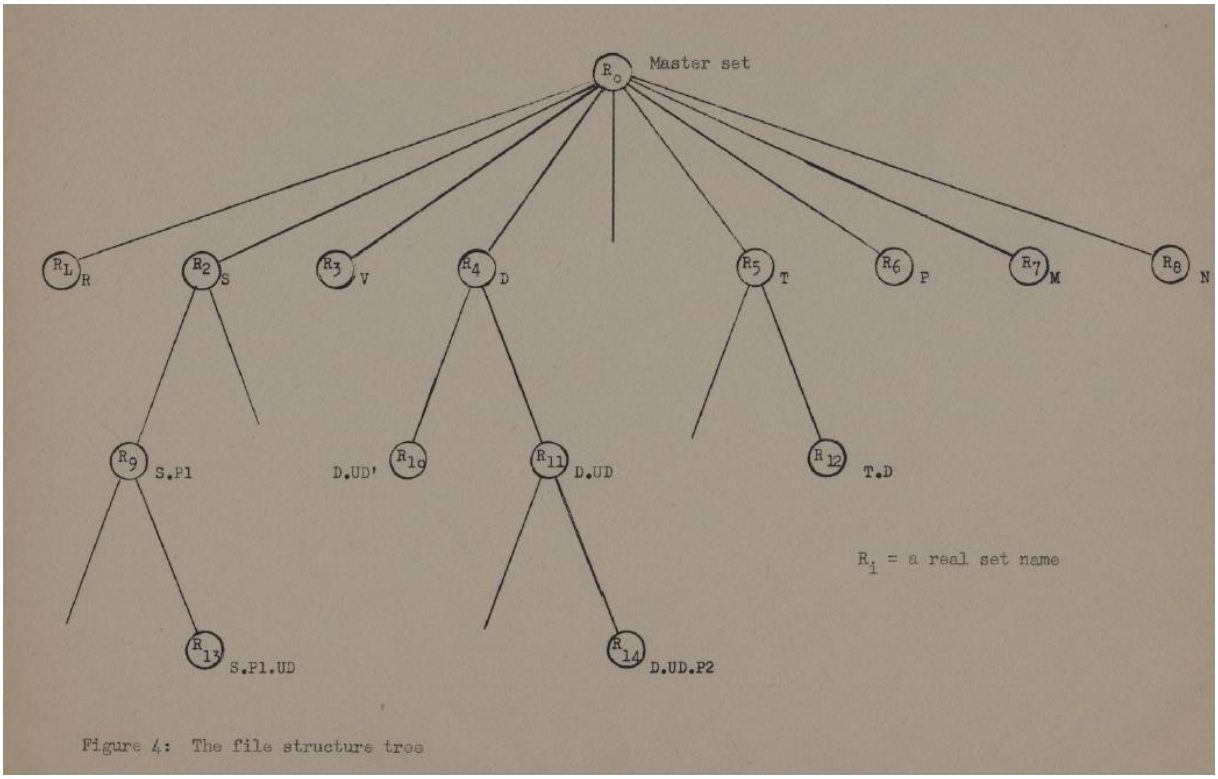

Figure 4: The file structure tree

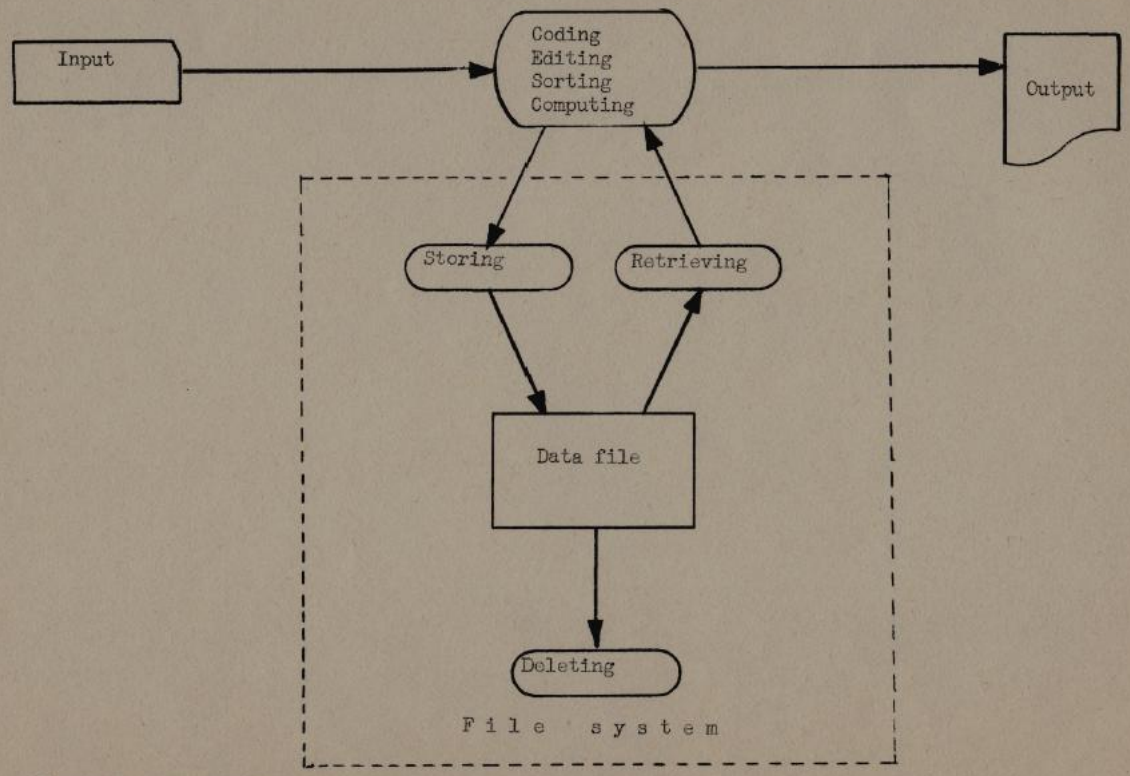

Figure 5: Statistical processing

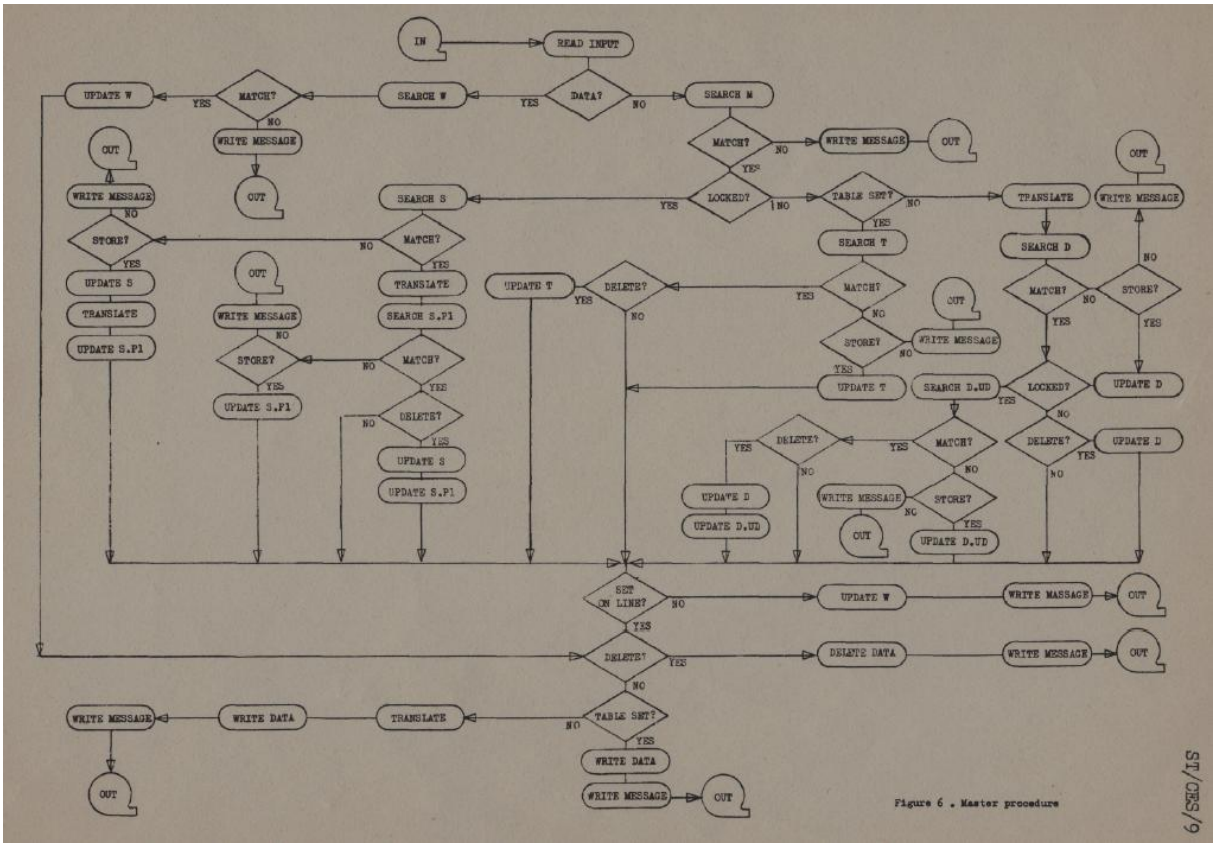

Figure 6 . Master procedure

ST/CES/9